3-24-2016

Analysis of Software Design Patterns in Human Cognitive Performance Experiments

Alexander C. Roosma

ANALYSIS OF SOFTWARE DESIGN PATTERNS FOR HUMAN COGNITIVE PERFORMANCE EXPERIMENTS

THESIS

Alexander C. Roosma, Captain, USAF

AFIT-ENG-MS-16-M-042

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-16-M-042

ANALYSIS OF SOFTWARE DESIGN PATTERNS FOR HUMAN COGNITIVE PERFORMANCE EXPERIMENTS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Alexander C. Roosma, BS

Captain, USAF

March 2016

AFIT-ENG-MS-16-M-042

ANALYSIS OF SOFTWARE DESIGN PATTERNS FOR HUMAN COGNITIVE PERFORMANCE EXPERIMENTS

Alexander C. Roosma, BS

Captain, USAF

Committee Membership:

Dr. Brett J. Borghetti
Chair

Maj. Christina F. Rusnock, PhD
Member

Dr. Leslie M. Blaha, PhD
Member

AFIT-ENG-MS-16-M-042

Abstract

As Air Force operations continue to move toward the use of more autonomous systems and more human-machine teaming in general, there is a corresponding need to swiftly evaluate systems with these capabilities. We support this development through software design improvements of the execution of human cognitive performance experiments. This thesis sought to answer the following two research questions addressing the core functionality that these experiments rely on for execution and analysis: 1) What data infrastructure software requirements are necessary to execute the experimental design of human cognitive performance experiments? 2) How effectively does a central data mediator design pattern meet the time-alignment requirements of human cognitive performance studies? To answer these questions, this research contributes an exploration of establishing design patterns to reduce the cost of conducting human cognitive performance studies. The activities included in this exploration were a method for requirements gathering, a meta-study of recent experiments, and a design pattern evaluation all focused on the experimental design domain.

Acknowledgments

This research results from the contributions of time and experience of many. To my faculty advisor, Dr. Brett Borghetti, who invested a substantial portion of his time guiding my writing and research, you have my sincerest appreciation. I also thank my committee members, Dr. Leslie Blaha and Maj Christina Rusnock, for their invaluable time investment and feedback, providing me insight into the world of human factors and cognitive performance research.

I am most grateful to my wife, for her unfailing empathy and words of encouragement. Most of all, I would like to humbly thank my God and creator, not only for creating a universe full of wonder, but also for the blessings and personal growth I would not have otherwise received through this experience. To Him be the glory.

Alexander C. Roosma

Table of Contents

Page

Abstract ..... **iv**
Acknowledgments ..... **v**
Table of Contents ..... **vi**
List of Figures ..... **x**
List of Tables ..... **xi**
List of Acronyms ..... **xii**
I. Introduction **1**
Key Concepts 5
Problem Statement 7
Research Focus and Research Questions 8
Motivation 9
General Approach and Thesis Overview 12
II. Literature Review **13**
Human Cognitive Performance Studies Background 13
Sources of Complexity in HCP Experimental Designs 20
Current Standards Development and Architecture Design 22
Summary 25
III. Methodology **26**
Data Infrastructure Module Taxonomy 26
Human Participant. *28*
Task Environment. *29*
Physiological Measures. *29*
Computational Module. *30*

Persistent Storage. *31*

Research Activities Overview 31

Data Infrastructure Queries 34

Meta-Study – Cognitive Performance Experimental Designs 39

Complexity Metric. *43*

Requirements Gathering 44

Inception *45*

Elicitation *46*

Elaboration *46*

Specification *46*

Validation *47*

Design Pattern Activities 47

Design Pattern Specification *48*

Design Pattern Evaluation *51*

Chapter Summary 53

IV. Results and Discussion 54

Data Infrastructure Queries Results 54

Meta-Study Results 55

Meta-Study Results *58*

Meta-Study Limitations *63*

Meta-Study Summary. *63*

Requirements Gathering Results 64

Elaboration *64*

Specification *64*

Validation *66*

Design Pattern Evaluation Results 67

Case Study 1 ..... *68*

Case Study 2 ..... *75*

CDM Analysis Summary. *77*

Discussion 78

V. Conclusions and Recommendations 80

Summary 80

Contributions 83

Future Work 83

Appendix A: Data Infrastructure Query Contents 85

Appendix B: Data Infrastructure Query Results 91

SME Interview 1 ..... *91*

SME Interview 2 ..... *92*

SME Interview 3 ..... *94*

SME Interview 4 ..... *96*

IRB Protocol Review 1 ..... *97*

IRB Protocol Review 2 ..... *99*

IRB Protocol Review 3 ..... *100*

Appendix C: Meta-Study 103

Publications Included in the Meta-Study 105

Meta-Study Coded Results 109

Bibliography 111

List of Figures

Page

Figure 1: Durkee *et al* data management architecture [3]..... 15

Figure 2: ACT-R Experimental Method [5] 16

Figure 3: Methodology Flowchart 32

Figure 4: CDM Communication Diagram 49

Figure 5: General Design – Central Data Mediator 51

Figure 6: Distribution of Surveyed Experiments (*n=51*)..... 57

Figure 7: Average Modules per Experiment over Time 59

Figure 8: Trend of Complexity over Time..... 60

Figure 9: Complexity Values With and Without Automation 61

Figure 10: Complexity Metric Grouped by Year Range 62

Figure 11: Basic HCP Experiment Use Case..... 64

Figure 12: Ad Hoc Module Concurrent Communication Diagram 69

Figure 13: Data Bus Module Concurrent Communication Diagram 70

Figure 14: Central Data Mediator Module Concurrent Communication Diagram..... 71

Figure 15: Concurrent Communication Diagram – SME Interview 4..... 76

List of Tables

Page

Table 1: Measures Used in Meta-Study 42

Table 2: CDM Evaluation Rubric 52

Table 3: Case Study 1 Evaluation Rubric 75

Table 4: Case Study 2 Evaluation Rubric 77

List of Acronyms

ACT-R..... Adaptive Control of Thought – Rational

CDM ..... Central Data Mediator

DOD..... Department of Defense

ECG.....Electrocardiogram

EEG..... Electroencephalogram

EMG..... Electromyogram

EOG .....Electrooculogram

EPICS.....Electrophysiologically Interactive Computer Systems

GSR..... Galvanic Skin Response

HCI.....Human-Computer Interaction

HCP ..... Human Cognitive Performance

HRV .....Heart Rate Variability

HUD.....Heads-up Display

IMPRINT .....Improved Performance Research Integration Tool

LOA ..... Levels of Automation

MATB..... Multi-Attribute Task Battery

NTP..... Network Time Protocol

OFS .....Operator Functional State

RPA.....Remote Piloted Aircraft

SADT .....Structured Analysis and Design Technique

SCL ..... Subjective Cognitive Load

SysML.....Systems Modeling Language

TCP.....Transmission Control Protocol

UAV..... Unmanned Aerial Vehicle

UDP..... User Datagram Protocol

UGV.....Unmanned Ground Vehicle

UML.....Unified Modeling Language

ANALYSIS OF SOFTWARE DESIGN PATTERNS FOR HUMAN COGNITIVE PERFORMANCE EXPERIMENTS

I. Introduction

To motivate and provide context for the research activities discussed in this thesis, we open with a hypothetical, representative operational scenario. The objective of this scenario is to provide concrete examples of the human cognitive performance products which this thesis seeks to support. The plot of this hypothetical scenario is a high value target extraction from a location in hostile territory. The friendly actors include an Unmanned Aerial Vehicle (UAV) video sensor operator, Unmanned Ground Vehicle (UGV) operator, and a squad of ground soldiers.

In this scenario, the UAV sensor operator has control over the aerial vehicle's video sensor pod to track two ground targets on foot simultaneously. At times, locating the targets within a complex environment can be extremely mentally demanding, such as when many similar looking individuals exist in the same visible area. This demand is mitigated by automated assistance of the locating and tracking of targets. The assistance is triggered at the most opportune time using an operator functional state model to continuously assess the cognitive load of the sensor operator. His cognitive load is based on a combination of physiological and behavioral metrics. These metrics include heart rate variability (HRV), eye movement, multiple electroencephalogram (EEG) channels,

and several key task performance features such as reaction time. To maintain optimum performance, when the operator's workload falls below a certain threshold, he is given additional manual control, and when workload has exceed a threshold, automated assistance is provided.

Meanwhile, a UGV operator maneuvers several ground vehicles towards the building while on alert for enemy contacts. The operator is able to achieve a higher level of performance during the operation because her task performance deficiencies were already identified and addressed in real-time during training. In addition to increased preparedness, a distraction model detects if the operator's attention drifts from the primary task using data feeds from an eye tracker and scenario updates.

Finally, a squad of ground soldiers equipped with Heads-up Displays (HUD) approaches the building. The message handlers for each soldier's HUD present a unique interface of operational intelligence and communication based on their cognitive state. Each of the soldier's cognitive state is derived from real-time data streams of multiple EEG channels identified as stable cognitive workload predictors during training sessions.

This scenario presents three types of real-time capabilities currently under development with significant potential for improvement to operations conducted by the Department of Defense (DoD). The first is the ability to assess the cognitive state of an operator and use that assessment to decide how to support their task to improve performance. The second capability is an improvement to training by providing automated real-time feedback that supplements or replaces trainers. The last portion of

the scenario demonstrates that this capability may also be employed in a mobile environment in a physical task.

The capabilities mentioned in the opening scenario and the methods of their application are created and matured through human cognitive performance experiments. If we support these experiments, then we can help create and develop these operational capabilities. Two ways that we can contribute to the experiments are by providing solutions for additional functionality and making it easier to achieve current levels of functionality within the execution of the experiments.

A major component of these capabilities is the ability to use real-time streams of physiological and behavioral sensor data from the human operator. The data streams flow between multiple computational processes and some also need to be time synchronized during the execution of the experimental designs. These types of communication create significant complexity that must be managed by nontrivial software. This software takes the form of a data infrastructure to provide researchers the ability to execute their experimental design. Finally, it is by providing a discussion and analysis of solutions to the data infrastructures that this thesis seeks to support these operational capabilities.

We define human cognitive performance studies as experiments that analyze the elicitation, manipulation, or observation of cognitive state changes in human subjects. Adaptive automation is defined as computational control over all or a portion of the task. The automation may be statically enabled or dynamically enabled by a trigger such as cognitive workload estimation.

Time-alignment is a key capability in the scenario we just presented as well as the execution of many other real-world experiments. Time-alignment is the time synchronization of two or more continuous streams of data. The synchronization is essential in this case because all the cognitive state models observe distinct physiological and behavioral responses from the same task stimuli.

One key element that must be time-aligned in human cognitive performance experiments are the physiological observations from the human subject. Physiological sensors can measure a range of physical features from the human subject, such as heart rate, skin conductivity, and pupil size. Psychophysiological observations are a subset of physiological observations that are characterized by neurological measurements of the human mind, such as an electroencephalogram.

While not all of the capabilities presented in the opening scenario currently exist at the level of functionality described, groundwork to deploy them is ongoing within the research community. Due to factors that include the decreasing cost of equipment, development of new technology, and focused effort by the Department of Defense (DoD), the quantity of experiments to support the types of operations included in the opening scenario is expected to increase.

Several key research activities are crucial to develop and refine the functions that appear in the opening scenario. One of these is the ability to derive a human operator's workload. A second is to understand how operators interact with automation and machine agents. Yet another is to understand how certain situational elements of a task scenario affect human cognitive resources.

Each of these research activities exist within the domain of human cognitive performance, but the software systems to make this research possible are not native to that same domain. They require development from a different domain of knowledge, software engineering. It is these systems which we seek to improve, specifically the data infrastructure enabling the flow of the data streams and their temporal alignment.

There are many individual software solutions to meet experimental designs throughout the human cognitive performance research community. The similarities in these solutions are due in part to a set of consistent functionality required to run experiments to answer research questions. Redundant design is not a unique phenomenon, and leveraging the patterns of these designs is possible. This idea is illustrated with the following often quoted analogy on the design of buildings and towns. Christopher Alexander wrote, “Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice” [1].

Key Concepts

There are a number of terms and concepts used in this thesis that are uncommon or have a unique meaning in the context of this research. In order to provide a common context, we provide the definitions of these terms referred to often throughout this research. Some are reiterated versions of commonly used definitions, while other definitions are novel to this research.

- Human cognitive performance (HCP) experiments: A categorical set of experiments that seek to further the understanding of cognitive functions in the human mind.
- Data stream: A continuous time-ordered series of data points. An example is the instantaneous measure of heart rate over time sampled at 5 times per second (5 Hz).
- Experiment data infrastructure: The hardware and software configuration collecting and routing data during the execution of an experiment.
- Experimental design: The design of a systematic task or set of tasks necessary to achieve the research goals. The design describes the conditions under which to run the experiment, the data to be collected and how the observed data is analyzed.
- Experimental design execution: The sequence of practical, concrete tasks that must occur during the execution of an experimental trial in order to collect the data necessary to achieve the research goals. The scope of this thesis covers the design execution phases starting with collecting observations, to processing the data if necessary and finally storing the results for post-experimental analysis. The analysis portion of the experimental design is not within the execution scope.
- Central Data Mediator (CDM): Software design pattern that describes how data streams and communication between components of an experiment are managed. More specifically, the design pattern provides a general architecture

and descriptions of communications between the data flow producing and consuming modules of an experiment. A software design pattern is a generalized, reusable solution for a common problem in a given domain.

- Software requirements: The functionality of a system required by the customer (cognitive performance researchers are the customer in the scope of this research).

Problem Statement

Data stream management requires a nontrivial solution when multiple distinct streams of data (e.g., EEG measures and task performance) must be synchronized and transmitted in real-time. Add to this challenge multiple versions of an adaptable task interface and maintaining alignment between the physiological sensor streams and task performance measures, and the necessary software infrastructure becomes even more complex. For these reasons, researchers in this domain require nontrivial software designs to execute their experimental designs.

The complexity necessary to execute the experimental designs of current human cognitive performance studies will only continue to grow and will include important new capabilities for adaptive automation and the understanding of human cognitive process in complex environments. In an effort to meet ever increasing demands of automation, the complexity of each of these experiments will also continue to grow. One factor of the complexity is time alignment of the data within the execution of the experiment. Time alignment is an essential activity for these researchers to correlate the observations from discrete sources within the experiment.

A simple example illustrating the necessity of this activity is the time alignment of presentation stimuli within an experiment to the physiological observations from the human. If one does not know when a psychophysiological measure occurred in relation to when the stimuli occurred, then little to no conclusions can be drawn from the data.

The ultimate vision for developing software patterns in this domain is to eliminate barriers of human cognitive performance experimental designs caused by software architecture limitations. It is very likely that no single tool could address the needs of the entire spectrum of experimental design configurations. Thus, our goal in this thesis is to provide groundwork towards this vision. Our focus is on real-time and augmented automation dynamic tasks because these are some of the primary drivers of complexity.

Research Focus and Research Questions

The research focus is software requirements and design. The scope of requirements in this thesis is the set that describes the flow of data critical to the execution of a human cognitive performance experimental design. Supporting data streams requires certain functionality from software, specifically from the software that serves as the data management infrastructure. The focus is on the common necessary characteristics of this experiment execution supporting software.

The process necessary to answer the research questions presented at the end of this section requires an understanding of the cognitive performance problem domain to include concepts such as: human factors, cognitive science, human-computer interaction, decision making and adaptive automation. We use a software engineering approach to investigate opportunities to both meet and provide solutions to make it easier to meet the

requirements of human cognitive performance experiments. From this software paradigm, we analyze three software designs that use different strategies to meet the data management to execute human cognitive performance experiments.

One of the research goals is to understand the root causes that drive an increase in software complexity within experiment software infrastructure solutions. To improve the field's ability to address challenges designing software that meets data management needs, a firm understanding of the challenges is necessary. The research focus reflects this goal of increasing the understanding of the software complexity challenges.

In order to begin building a foundation to solve the problems that face human cognitive performance experiments, this thesis answers two research questions addressing the data stream management software infrastructure necessary to execute the desired experimental design. The first question focused on gathering requirements of human cognitive performance experiment design. The second question focused on the evaluation of a specific design approach in the context of the requirements.

- *What data infrastructure software requirements are necessary to achieve the research goals of human cognitive performance experiments?*
- *How effectively does a central data mediator design pattern meet the time-alignment requirements of human cognitive performance studies?*

Motivation

The preconditions for performing analysis of the results of a human cognitive performance study are the ability to collect continuous, dynamic time-series of

observable measurements, using those data streams to inform models and make decisions, and storing the data for later analysis. Managing the data produced and processed during the execution of one of these studies is often a nontrivial exercise due to the large number of experimental components, the difference in functions performed by each component and the magnitude of data generated. The experimental modules may include: 1) human subject physiological sensors, 2) the simulated task environment, 3) persistent storage mechanisms, 4) computational models and 5) adaptive automation controllers.

There is no single best feature provided by physiological measurements for all cognitive manipulations (factors and levels) of a task [2]. A large quantity of data is the result of high resolution physiological observations of the human subject such as EEG. These types of measurement may produce multiple channels of data with sampling rates over 1,000,000 samples per second or 1 Megahertz (MHz). Physiological data in these experiments is primarily used as an indicator of the cognitive state of a human subject. Physiological measurements are not the only source of high-fidelity data streams during the experiments.

A non-physiological source of high-density data may be a simulated task environment. A continuous stream of task performance metrics is one common data stream from a task environment. The environment state representation for the task shared with other components is another possible output. Maintaining an accurate record of the task state throughout the trial is essential for analysis, so that the human subject responses can be time-aligned with the event stimuli and physiological sensor data.

In this research, we are interested in the portion of the experiment that manages the collection of data from every producer and consumer of data within the experiment. To clarify, the scope of this research does not include any study of the actual values of data from a study, but rather the process used to manage the flow of the data.

The potential number of different software solutions to the problem of providing data management is staggering. There are more experiment data management software architectures than there are potential human-machine team performance studies. However, there are many common design aspects between all of the studies which can be leveraged to reduce the complexity of the data management architecture design and implementation.

In order to provide context for Chapters 2 and 3, we briefly list below the requirement statement results from Chapter 4. The common data requirements gathered from published experiments and queries of data architectures by this research are as follows:

- Data streams from disparate processes not operating on the same system clock shall be time-series aligned.
- Modules within the experiment shall have the ability to receive multiple streams of data from one or more other modules in real-time.
- Modules within the experiment shall have the ability to send multiple streams of data to one or more other modules in real-time.
- Multiple data types shall be handled simultaneously. Examples are double floating points, integers, strings, binary, images and video.

- The available physiological hardware sample rates shall be maintained.

General Approach and Thesis Overview

There are many possible approaches to improve the research methods of designing human cognitive performance experiments. However, this thesis focuses on one: improvements to developing and integrating the data management software infrastructure necessary to execute an experimental design.

The thesis is divided into five chapters. I. Chapter II provides further details and challenges of human cognitive performance experiments. These include motivation, core concepts, current challenges and what is being done to address those challenges. Chapter III presents the methods used to answer the research questions. This includes novel constructs necessary to perform the. Chapter IV presents the results of executing the methods to answer the research questions. The conclusion, in chapter V, contains a distillation of the results and the lessons learned from the methods used. The operational significance of the results is discussed next and, recommendations for future work following the results of this research are presented.

II. Literature Review

This chapter provides the context necessary to discuss the data infrastructure requirements of human cognitive performance experiments. While software tools that manage the complex communication of data streams have been developed in recent years, there has been no theoretical discussion of the requirements that these tools are built to support. Further, while there were several examples of software product solutions, discussion on general software approaches and design were not seen in the literature. We discuss this concept using recent published works in human cognition research.

We build a foundation of knowledge on human cognitive performance experiments by focusing on several domains in the research literature. The first domain is background on various research goals within human cognitive performance studies. Second, we present literature that represents human cognitive performance research including or relying on complex system interactions. Finally, current and past approaches for creating standard interfaces for the components of a human cognitive performance experiments are enumerated.

Human Cognitive Performance Studies Background

The following works are a primer on the types of research activities conducted within the domain of human cognition performance experiments. Many research efforts in the human cognitive performance domain seek to answer questions such as: What are the physiological markers of cognitive state from external stimuli? How can that

information be employed to inform a computational model to take optimal actions in concert with a human actor? A common measure used to answer this question is the operator's cognitive workload or functional state. Due to the vast number of uncontrollable, directly unobservable, and unrecognized factors that play a role in an operator's cognitive workload, many distinct experiments are necessary to expand the current body of knowledge. Additionally, the interactions between human and computational agents are dynamic. The large quantity of research necessary to forward the field suggests the necessity of an architecture to improve the ease and capability of performing the research.

Durkee *et al* addressed the question of how to identify when to apply adaptive augmentation, and to answer this, focused on the ability to measure and predict workload in real-time [3]. The authors presented a data aggregation and modeling architecture to describe their approach of modeling workload in real-time. The model, shown in Figure 1 below, serves as a source providing a listing of components that exist within the experiment's data infrastructure. The diagram also demonstrates how the components and the data streams flowing between them may be arranged. The data bus serves as one example of a standard interface to improve modularization and extensibility.

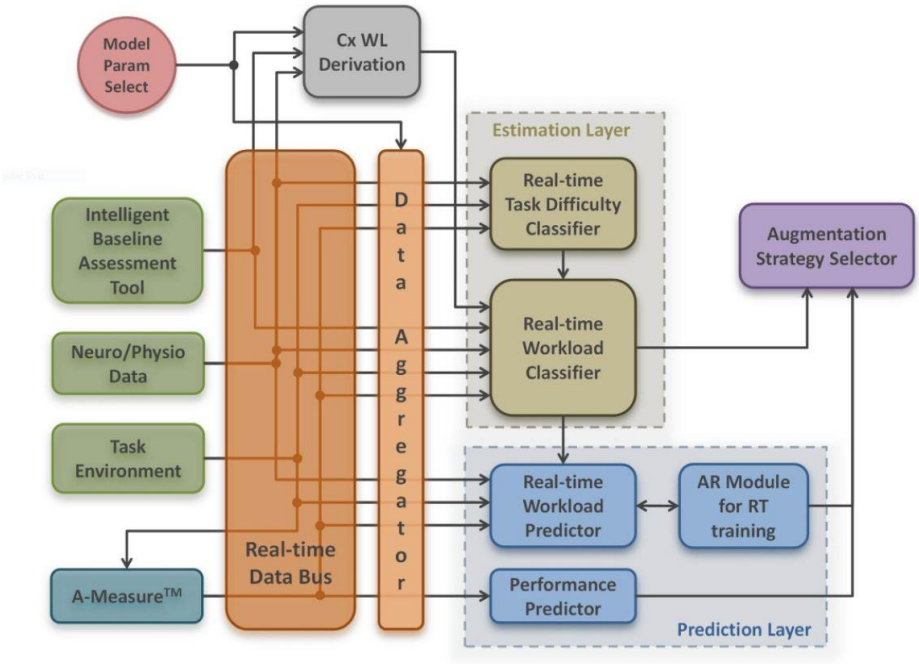

Figure 1: Durkee *et al* data management architecture [3]

The authors also presented their application of adaptive automation as a tool to modify the user interface of a task to match the inherently dynamic states of the operator. As a motivator of using physiological measures within their model, the authors point to the architecture's proven effective use as a predictor of workload. Three insightful limitations observed by the authors regarding current approaches to measuring operator states provided were: 1) reasonable model accuracy was not possible without "training" to a specific individual, 2) temporal gap between observation and assessment, not being able to derive a classification in real time, 3) overall lack of granularity and timeliness of operator state assessments.

One example of enabling communication between disparate models from different cognitive modelling paradigms is presented by Lebiere *et al.* The authors

presented work on integrating IMPRINT and ACT-R that focuses on demonstrating the possibility of aligning these disparate computational models in such a way that they exchange information on their respective levels of task execution [4]. Figure 2 depicts a generic demonstration of ACT-R's role with an experiment. The complexity of the task is twofold: one, to decompose the task appropriately, and two, and probably more difficult, to pass the necessary information between the models in a format that can be ingested into each. The communication is accomplished in a narrow scope, only engineering communication between the two specific instances of models they explored. The authors did not, however, discuss extending the mode of communication to a standard format that could be applied to other forms of models or even other types of components, such as synthetic task environments (STEs) or physiological measurement devices.

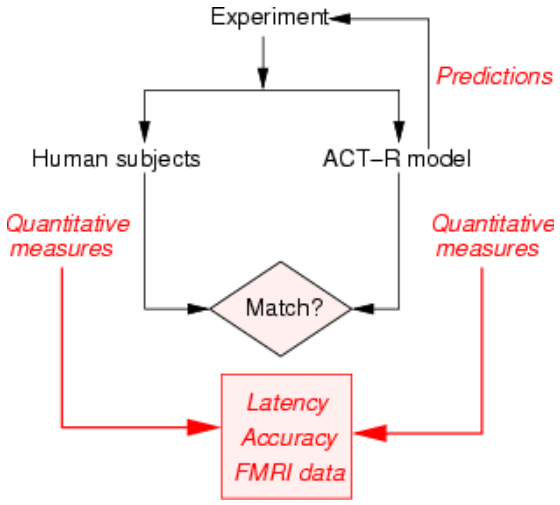

Figure 2: ACT-R Experimental Method [5]

Additionally, Lebiere *et al* ran simulations with the models rather than having the cognitive models interact with the same instance of the task environment as a human

subject. The authors note the act of sharing variables, such as clock time, between models is an essential requirement of conducting the experimental design. Variable sharing can be extended to additional components of the experiment design beyond just the models. The large amount of information sharing necessary between the models suggests that message passing between parallel computational models in an experiment is a workflow requirement.

Allanson and Fairclough's literature survey focuses on the argument for standard processes, software libraries, and architectures of human computer interaction systems which integrate physiological measurements [6]. Allanson and Fairclough also include a robust listing and description of detectable human physiology measures which include: EEG, electromyogram (EMG), electrooculogram (EOG), pupillometry, electrocardiogram (ECG), respiratory patterns, electrodermal activity/galvanic skin response (GSR), and blood pressure. Two requirements noted by the authors for systems that integrate physiological measures are 1) the necessity to retrieve physiological data from an external measurement module, and 2) the necessity to have some degree of pre-processing of the physiological data suitable for consumption by edge components that employ the data. One potential solution to accomplish pre-processing proposed by the authors is to use tuned thresholds to determine if the raw data indicates a real physiological response. Other physiological data streams may require filtering on specific bandwidths via one of several spatial filtering methods.

While the authors express passionate support of standardized development practices, they do not provide a full workflow perspective to include testing and research

activities. Following are two examples of activities called out by the authors as future work that need to be developed. The first is the need to develop algorithms with the ability to categorize physiological states and distinguish levels within those states from a basis of research. The second is the development of the ability to match a proper system response to a physiological pattern recognition, whether there is a significant or almost imperceptibly small change.

In a similar research effort, Allanson's PhD thesis provides additional, more in-depth discussions on the implementation details of physiological measurement devices as well as tools to aid in the design of human computer interaction (HCI) systems [7]. The author goes into detail over using conceptual models of component interaction to improve the design of systems that rely on communication between a human and a machine. The author also introduces the concept of Electrophysiologically Interactive Computer Systems (EPICS) as a class of systems that leverages human physiological measurements within the machine to make programmatic decisions.

Jo, Myung, and Yoon also focus on cognitive workload prediction capability; however, their work uses a tool, ACT-R, that does not natively provide cognitive workload prediction [8]. The authors suggest enabling workload prediction in non-native tools can be extended to other models and thus this is something that should be supported by the framework. A list of cognitive architectures compiled by the authors includes: ACT-R, Executive Process/Interactive Control (EPIC), Soar, and Queuing Network – Model Human Processor. The capability presented in Jo *et al*'s work is limited to the domains supported by the ACT-R tool, which excludes certain aspects of human

cognition such as vigilance tasks and mental fatigue. Similar to many experiments in the human cognitive performance domain, the dependent variables measured by the authors were task completion times and subjective workload, NASA Task Load Index (TLX).

Mancuso *et al* discuss their human team performance study, investigating differences in integrated and differentiated knowledge structures within distributed teams [9]. The authors' primary goal is to investigate whether common knowledge or a disparate, wider knowledge base can better improve human team performance. The environment used in this study is referred to as a scaled world simulation. The task is cyber-domain specific as it represents network intrusion detection analyst actions.

The task environment in Mancuso's study is more complex than many of the other human cognition experiments in literature which usually leverage a very simple task to reduce confounding factors. One possible future extension to this study using an implementation of the proposed framework would be to study the effects of various augmentation agents mixed into the teams. This team effectiveness study suggests that a framework design should consider the inherent or extensible ability to support experiments with multiple task environment instances.

Subjective mental workload measures are another important measure in human cognitive performance experiments as Wiebe *et al* discuss [10]. Specifically, the authors covered the following subjective workload assessments: NASA TLX and Paas' Subjective Cognitive Load (SCL) measure. The authors point to Eggemeier *et* al's work which classifies mental workload measurements into three categories of 1) subjective self-assessment methods, 2) task performance measures, and 3) physiological indirect

measures of workload [11]. The implementation of their methodology to study the differences used a visual diagram whose design was altered to manipulate the independent variables.

Wiebe *et al* also describe several components of Cognitive Load Theory with respect to learning, namely intrinsic load, germane load, and extraneous load [10]. These concepts may be used with respect to the framework in extending its application to the learning and instruction domain beyond HCI and workload theory studies. The study also raises the question of how the framework might integrate manually collected subjective data into the framework, especially if the researchers choose not to use a computer based worksheet or questionnaire.

Sources of Complexity in HCP Experimental Designs

This section of the literature review uses existing research to provide real-world and theory-based examples of the need for a software infrastructure design in human cognitive performance experiments. Some of the examples are extensions of existing systems, while others are proposals for novel activities that rely upon non-existing capabilities that would be provided by this framework.

A new human cognitive performance assessment system developed by the Air Force Research Laboratory called the 24/7 Combat Fitness System is slated for release in 2016 [12]. The common theme between this system and this thesis is the assessment and prediction of human cognitive performance. The goals and outputs of the system provide further evidence of the need for human cognitive performance research in the near future and beyond. In the article, Dr. Scott Galster, chief of the branch developing the system,

explains that the goal of the 24/7 Combat Fitness System is to provide leaders with the capability to assess whether their team is performing optimally as well as understanding the factors that lead to their team's success.

The system performs many of its tasks in real-time and gathers data continuously to provide immediate assessments. Real-time assessment necessitates a shift toward on-line model usage rather than post hoc analysis. An increase in real-time assessment and prediction will require a corresponding increase in research to support these activities. The research would benefit from a standardized approach to the experiment communication design in order to conduct a greater quantity of studies as well as adding the ability to assess cognitive model performance and make experimental level adjustments in real-time. These are outcomes that this research of the proposed framework is building toward.

Bindewald *et al* provide evidence of experiments with high degrees of data infrastructure complexity, as the authors describe a human cognitive performance experiment employing adaptive automation [13]. The sources of complexity are the numerous modes with which a human and machine can dynamically balance task load between each other. The authors note the large degree of information that needs to be shared in real-time between the human and the computational machine providing automation. This increase in information sharing, especially with a real-time stipulation, increases the demand of functionality of an experiment's data management infrastructure. Tools mentioned in the study for capturing interface design requirements include Systems Modeling Language (SysML), Structured Analysis and Design Technique (SADT), and

Unified Modeling Language (UML). These tools may be applicable to other human cognitive performance experiments. Bindewald *et al* provided a walkthrough of decomposing a high-level human function to distinct tasks that could be allocated appropriately between the human and task automation.

One additional research effort that provided motivation of improved human computer interfaces and adaptive automation and thus the research and experiments that support them was conducted by Kaber *et al* [14]. Their work discusses the paradigm of considering the human within the experiment as another module of data processing. Many of the implementation issues the authors present are centered on challenges in the communication of information between human subjects and machine systems. The common characteristic indicates a corresponding need to manage the flow of information to support various methods of presentation.

Current Standards Development and Architecture Design

This last section of the literature review research presents existing efforts in connecting components of the human cognitive performance experiment data architecture.

Research conducted by Halverson, Reynolds, and Blaha represents work dedicated to improving researchers ability to construct the equipment necessary to execute their experimental design [15]. The scope of their research is focused on the task environment and cognitive architecture model components of an experiment. A key concept broached by the authors with respect to a standardized experiment architecture is

that there should be minimal modification or re-configuration required of the original task source code.

Halverson, Reynolds, and Blaha present several arguments in support of using a software framework rather than re-implementing a very similar task design to study a similar interface or cognitive task. The overall theme for these reasons is that re-implementation requires time resources that do not benefit the study which is the purpose for building the experimental software architecture.

Cohen *et al* focus on abstracting aspects of the cognitive model development process [16]. This is another example of a research study aimed at enabling researchers to abstract from implementation configurations and details. The primary goal of the language presented is to enable the creation of models that contain explanations of their design and is based on developing models in the Soar language. The authors establish a standard representation by using an ontology to describe the relationship between the high level classes.

Cohen *et al* use a sample implementation of a Soar model to aid in describing their language. The main advantage noted by the student participants in the Cohen's study was implementation reuse in the form of conditions and actions for the Soar model. The authors include strategies for connecting modules of a study that serve as both producers and consumers of data and actions. These strategies include creating an ontology, simplification of cognitive model development, and generation of explanations for running models.

Following are some examples of cognitive performance that is not directly cognitive workload, that show other measures of human cognitive performance that are studied. The importance of this is that a general framework for human cognitive performance experiments should be applicable to the multiple cognitive performance experimental designs currently executed in this field. Another cognitive architecture that contains elements of a standard framework is EPIC (Executive Process-Interactive Control) [17]. One of the significant contributions of validated models from this architecture is the ability to drive the design of HCI systems. One example of a human subjects experiment not directly on cognitive workload, but a related concept, situation awareness, is detailed in research by Giacobe [18]. Giacobe presents a study on situation awareness and cognitive measures in simulations using methods not as commonly used as the rest of the literature. While the standard NASA TLX was used to measure subjective workload, a situation awareness assessment was conducted via a short quiz during breaks in the task. The Sense-Assess-Augment taxonomy produced by AFRL is yet more research that describes standardized communication techniques [19].

One of the most comprehensive products for providing a solution to managing the many potential components of a human cognitive performance experiments is the Fusion High Level Framework [20]. Rowe, Spriggs, and Hooper present a novel framework that meets a wide range of functional requirements to include integration of real-time models, managing control and data message passing and presenting a standard user interface the human participant. Their solution also employs an application programming interface

(API), which provides a single, standard interface for new components added to the experiment.

Summary

The literature review first described the makeup of the experiments for this research. Next, several of the core, common data architecture designs were discussed. Lastly, this chapter illustrated the operational motivation for improving these studies. In the next chapter, we transition from introducing the varieties and components of human cognitive performance experiments, to the methodology used to identify and analyze properties of those experiments.

III. Methodology

This chapter builds on the familiarization with human cognitive performance experiments and their challenges to introduce methods for analyzing and addressing the complexity of their data infrastructures. This focus on the experimental software data infrastructure includes identifying common experimental designs across published studies and the practical challenges in achieving those designs. The following sections describe how the data were collected, analyzed, and presented to answer the research questions. Answering each research question required distinct and, at times, novel methods.

First we introduce a taxonomy that is used in each of the activities to describe key elements of the data infrastructure. Following the taxonomy description, an overview of the four research activities provides a consolidated description of all the planned activities. Finally, the details of each activity are presented to describe the actual process and contents.

Data Infrastructure Module Taxonomy

The elemental building blocks of data management software infrastructures need to be defined in order to clearly discuss an infrastructure's configuration and requirements. The elements are defined to create a standard way of describing the components so that the same definitions apply across a wide range of human cognitive performance experiments. Since such a construct was not observed in the literature, we developed the following taxonomy. This section first discusses overviews categories and boundaries of the taxonomy and then lists the specific instances of classification.

In order to provide context for the introduction to the taxonomy, the categories of the taxonomy are listed here.

- Human Participant
- Task Environment
- Physiological Measure
- Computational Module
    - Computational Agent
    - Computational Analyzer
- Persistent Storage

The scope of experiments covered was limited to those that offer potential benefits to DoD operations. One of the sets of operators supported are Remote Piloted Aircraft (RPA) operators with the Multi-Attribute Task Battery (MATB) task. Another group is cyber operators, such as through studies on team dynamics in a cyber task [21]. Tactical planners are also supported through efforts such as the Fusion framework and the studies that use the design [20]. The specific experimental components that each of the analyzed studies have in common are first listed and then described in further detail. Each includes a human participant, a physical or computer simulated task, one or more streams of physiological or behavioral measurements collected from the human participant, zero or more computational agents, and finally, persistent storage.

The data-producing and data-consuming components of an experiment are defined according to their function in this taxonomy. The taxonomy is used to orient the discussion on measurements of complexity and requirements within an experiment's

software configuration. The boundary of each module is defined as the point when data produced by one module is sent to another process that does not share the same computing memory. This means that the data must be intentionally packaged in order for the other process to use it rather than passing a pointer to the data in memory. Processing in this case could also include persistent storage for later analysis.

One specific example of defining a module boundary is considering all EEG output collected from a single piece of hardware a single module, as opposed to an individual module for each channel of the signal. Multiple channels will likely be collected off a subject's scalp, which are then passed in analog format through an amplifier to a digital converter. All processing up to the point where the data stream is digitized is considered part of a single module, as opposed to an alternative, considering that same configuration two modules. This alternative is one module for the analog signals collected off the scalp, and a second that consumes the voltages from the scalp and outputs a digitized time series. Therefore, the digitized data distinction is used to abstract away unmodifiable hardware aspects of physiological collection from module and infrastructure configuration.

Human Participant.

There are at least two cases of the human subject's participation in this research's scope of experiments. The human participant may be the primary subject of the study, and if they are not then it is likely the case that one or more of the machine agents are the focus of the study. The human in the second scenario is usually included in the experiment as a comparison to the computational models as verification or validation.

Task Environment.

A task environment is a system that provides a stimulus presented to the acting agents within the experiment whether those agents are human or computational, machine agents. The spectrum of tasks for which the research questions apply covers two dimensions. The first is synthetic vs real-world environments. Synthetic environments are those that exist purely within the confines of a computational model. One example is a Synthetic Task Environment (STE) that is a computational environment that also takes in real-world information. Another type of synthetic task is a fully computational task that can be a simplification of a real-world task or a task designed to elicit specific cognitive and physiological responses. The other end of that dimension are those which exist only in the physical domain. The physical end of the task type spectrum are those where all of the environment operations are completely within the physical domain.

The second dimension is a physical vs mental action of the human subject. For the physical dimension, the task type is a physical action such as running or performing a physical maneuver. The other end of the spectrum, mental task types are those where the task is performed in the mind such as addition or memory tasks.

Physiological Measures.

Detectable human physiology has been studied for over 70 years, increasingly so to understand the responses to various psychological and physical conditions [6]. The observations originate from specialized hardware designed to measure discernable signals from a human subject. Several examples are heart rate, pupil dilation size, and respiratory patterns. A subset of physiological measures are psychophysiological

observations which refer to those measurements made on the cognitive portion of the human, such as EEG.

Computational Module.

A computational module, which has two sub components, is defined generally as a module that takes an input, performs some algorithmic computation, and outputs a result. The two subcomponents differ on the type of output, information or action. The computational agent is the module that outputs an action, while a computational informational unit outputs information that is not a direct task action.

Computational Agents.

A computational agent is an instance of a computer process that performs functions on input of data and stimuli within the experiment to produce an output. The general form of these agents is: Sense, Decide, Act. The inputs, Sense, are the perceptual inputs of the agent to the world. Decide corresponds to the algorithmic processing of the data. Finally, Act is the resulting action to be carried out within the task environment [22].

Computational Analyzer.

This type of module performs some type of computation on inputs of data from within the experiment. The output may include information such as Operator Functional State in terms of cognitive workload, a specific level of automation to employ, or modeled neurological signals.

Persistent Storage.

This is storage that is used to record observations from each trial of the experiment. The primary function of the storage is to maintain the data for analysis after the completion of the experimental trials.

Research Activities Overview

Four primary research activities were conducted to answer both of the research questions. All of these activities are depicted in Figure 3. Each activity is separated by a horizontal dashed line. The horizontal groups can be read left to right as that activity was performed with the pictured resource to result in the listed products. The arrows indicate that the output of one activity was used as a resource for another. The timing of the activities may generally be read left to right, top to bottom; however, it is not a strict ordering as the activities were accomplished concurrently to an extent. The activities are: 1) *data infrastructure queries*, 2) *meta-study*, 3) *requirements gathering*, and 4) *design specification and evaluation*.

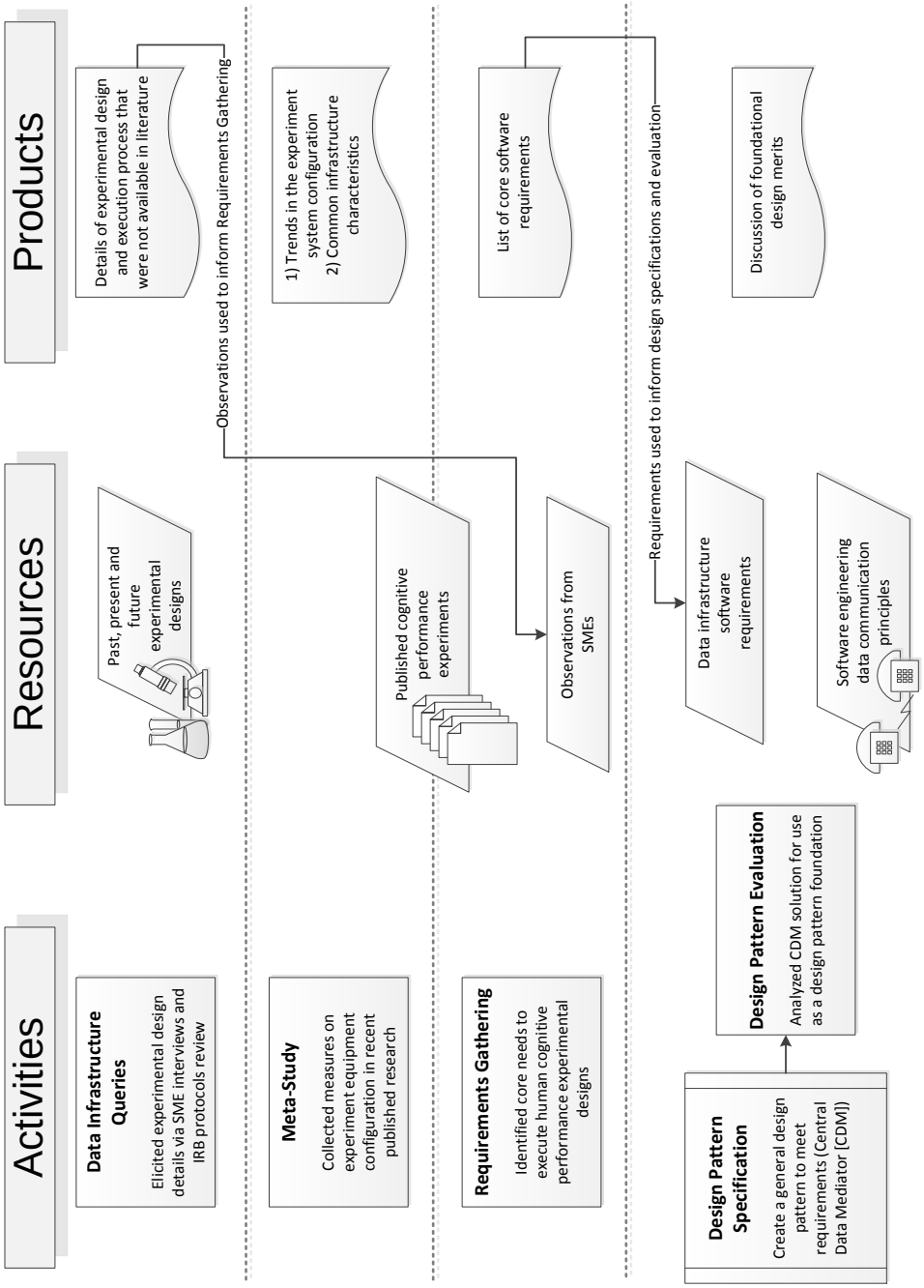

Figure 3: Methodology Flowchart

The purpose for the first activity, *data infrastructure queries*, was to observe details not otherwise available in published research. These details were then used to inform both the software requirements to help answer the first research question and the design pattern construction to help answer the second research question. This activity also provided an opportunity to anecdotally capture relevant challenges encountered by the human cognitive performance researchers.

The data infrastructure query activity consisted of eliciting experimental design details from subject matter experts (SMEs) and Institutional Review Board (IRB) protocols. The SMEs were research scientists of the Air Force Research Laboratory who conduct experiments in the human cognitive performance domain. The SME interviews collected data on details of the equipment configuration of planned or completed experiments (no personal information was collected). The data collected were mostly of a qualitative nature because the focus was on describing the configuration of a system.

The second activity, a *meta-study*, was conducted in order to achieve a broader view of the state of human cognitive performance experiments. Through a review of a sample of studies from the global population, we could gain evidence to reason for the larger population of human cognitive performance experiments. Requirements gathering, the third activity, was conducted in order that the proposed design pattern and future designs would be a resource for drawing design choices from by software developers.

The fourth activity was the *design pattern evaluation*. This activity consisted of two parts, the first of which was to create a notional design pattern specification to review. The second part was an evaluation of the design using a case study.

Data Infrastructure Queries

The following section presents the motivation, contents, and execution plan of the data infrastructure query research activity. This section first discusses why this activity was a chosen method to answer the research questions. Next, the contents of the questionnaire used for the SME interviews and IRB protocol reviews are discussed. Finally, considerations for how the activity was conducted are discussed.

There was not enough detail in published human cognitive performance studies literature to capture all of the software requirements. Through the literature review in Chapter 2, we observed that the majority of methodology sections contained little or no description of how the components in the experiment were connected. The methodology sections lacked information describing how the data streams were managed during the experiment. An example of desired information not available in the published research is a description of the data-passing architecture. This architecture is part of the data infrastructure, specifying how each of the distinct modules are connected to each other during the execution of an experiment.

Additionally, interviewing SMEs would provide a source for collecting anecdotal evidence of common challenges in executing their experimental designs. While the same challenges may not be discussed outright in published experiments, it is possible to infer the challenges from the experimental designs. The anecdotal observations could then be validated against observations from the wider body of published experiments.

In order to capture this information that was not available in published literature we queried primary sources on experimental designs. Those sources were the researchers

conducting human cognitive performance experiments and experimental designs. The product used to gather the selected measures from each source was a questionnaire. That questionnaire appears in Appendix A: Data Infrastructure Query.

The questionnaire to accomplish the data infrastructure queries was designed to gain insight into two characteristics of human cognitive performance experiments. One desired insight was the actual data infrastructure software solutions and configurations that researchers are currently using to execute their experimental designs. The second insight was to gain further details and clarification on the goals for the experimental design. The common goal of gathering these insights was to better understand the data management needs for the experimental design and how they are currently being met. Understanding the data management needs could then be used to make better suited design recommendations.

The following measures were chosen to capture relevant and measurable characteristics of the data management software infrastructure designs supporting the experimental design execution. The two general categories of measures collected are data infrastructure system complexity and experiment data management processes. The chosen system complexity attributes were:

- Module coupling
- Module cohesion
- Message communication growth
- Module count

Experimental data management process observations included:

- Post-Trial analysis
- Data collection and processing

The following sub-sections describe each of the measures to be collected and why they were chosen. The first several measures originate from software engineering metrics for describing the complexity of multiple module systems. The paradigm of what is defined as a module used in this section is according to the taxonomy at the beginning of the chapter. This definition of a module differs from the typical definition in software engineering which is a set of lines of program code.

Module Coupling.

Module coupling is the degree to which distinct components are interdependent on the configuration of each other, a measure of interconnectedness. The consequence of higher degrees of coupling is an increase in the work required to modify or exchange any single module. Since modules with a high level of coupling are very interdependent on the details of their connection, a modification of one will also require modification of the other. Alternatively, a very low level of coupling, such as a standard interface, enables modification, or even exchange, of modules to be hidden or abstracted behind the interface.

The responses to questions about this metric were used to assess where along the coupling spectrum the specific study's data architecture lies. A common theme of a high degree of coupling between modules of an experiment would suggest that an effort to provide a higher level of abstraction (encapsulation of the details that change) will have a positive effect. A lower degree of coupling allows the software components to be more

modular, requiring much less effort each time a new component is added or created for the execution of the experimental design.

Module Cohesion.

A similar and related concept, module cohesion, is a measure of the organization or centralization of the logic used to manage the data produced and consumed by experimental modules in the course of trials. The management of experimental data during execution consists of distribution to the applicable modules and collection for post-trial analysis. Cohesion is used to assess the quality of modularization which has been measured by the cohesiveness of service provided by an individual module [23].

Message Communication Growth.

The common measures of algorithmic space and time complexity were not used because they measure software execution at a lower layer of abstraction than the scope of this thesis. Inter-module interactions take place on a higher level of abstraction than the algorithmic design. A similar measure that may still be applicable within the scope is message communication growth. This refers to the growth of the total number of message packets that are required as the number of components, (n), in the experiment is increased. An architecture in which every component is connected to every other component, could have a message growth rate as large as n^2.

This measure can reveal complexity in an architecture that is not designed to efficiently handle the addition of many experimental modules that both produce and consume data streams, such as computational cognitive models (workload, vigilance, distraction measures) based on physiological data. The existence of a large growth rate in

an architecture may not represent capability-limiting impact at the moment; however, as research into building and testing multiple computational models and agents increases, the growth rate will begin to create limitations.

Post-Trial Analysis.

One of the descriptive observations collected was the post-trial analysis conducted after the runs of the experiment were completed. This observation consisted of a list of investigative questions from the experimental design and the techniques used to analyze the data. The primary purpose for gathering information on post-trial analysis was to find solutions to address challenges that occurred after the execution of the experiment before they became a problem. The purpose of gathering this list was to provide context to the rest of the experimental workflow. Another was to be used as a source to derive the requirements for the format of the data.

Module Count.

The simple number and type of modules was captured in order to provide a simple estimate of the size and complexity of the experimental architecture. There are other aspects such as whether the modules capture data in real-time or the task environment has dynamic levels of automation that affect the communication complexity just as much, if not more than the number of modules. The modules are defined and counted according to the taxonomy at the beginning of this chapter. The end goal for this measure is to supply a summary statistic that is on the same scale across both observed experiments and literature survey experiments.

Data Collection and Processing Limitations.

A listing of data collection and processing limitations was used to gain insight into any requirements for the design of the architecture that were not met, and thus could not be inferred from an external analysis of the existing architecture. The scope of the limitations applied to both the experimental trial execution phase as well as the post-trial analysis phase.

The primary sources for this information were in-person process description elicitation sessions with active researchers in the field. The process description elicitation sessions consisted of questions about the experimental workflow design and were used to elicit information that was not otherwise available in documentation. In addition to questions in person, review of other available design products, such as experimental protocols, was conducted.

Meta-Study – Cognitive Performance Experimental Designs.

In order to gather evidence to describe the current state of data management complexity in human cognitive performance experiments, we reviewed published papers in the format of a meta-study. The purpose of the meta-study was to investigate potential trends and common characteristics of the experiments. This purpose was accomplished by eliciting specific characteristics of published experiments that indicated the data infrastructure architecture and configuration. The papers reviewed were published between 1996 and 2015.

To perform the meta-study, we borrow the methodology format from the meta-analysis. A meta-analysis differs from this meta-study in that the contents, a common

statistical measure, presented in each study is not the target measure, but rather the setup, information about the experimental design. The five steps in a meta-analysis are: 1) Formulation of the problem, 2) Search of literature, 3) Selection of studies ("incorporation criteria"), 4) Decide on dependent variables, and 5) Selection of a meta-regression statistical model.

First, the problem to answer was an investigative question that would improve our understanding of the current and, especially, near future requirements for the designs of experiments within the thesis's scope. The goal is to identify the current state of data management complexity in order to assess the costs and benefits of using a specific data software infrastructure are to achieve these experimental designs. The idea is summed up in the following conjecture: "There exists a trend in the increase in complexity of the communication between data streaming modules during the execution of human cognitive performance experiments." The trend may be due to several factors which include an increase in the need for real-time data stream processing, an increase in the fidelity and dynamism of task levels of automation, and a decrease in the cost of physiological measurement hardware.

Second, the literature search was conducted primarily using two sources of pre-collected publications. One source was a PhD candidate's literature review of adaptive automation. Another was a research group's collection of human-machine teaming publications. Additionally, one journal, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, was searched by keyword. The keywords used to

search these journals were: “real-time”, “physiological measures”, “operator state assessment” and “adaptive automation”.

Third, the incorporation criteria were those publications that included human cognitive performance experiments. Specific requirements for those experiments were: Each included a human participant, a physical or computer simulated task, one or more streams of physiological or behavioral measurements collected from the human participant, zero or more computational agents, and finally, persistent storage.

Fourth, we discuss the measures collected from each of the studies surveyed. Each of these measures are listed in Table 1 is described in the following paragraphs. Year published is self-evident and is collected in order to group and order the measurements temporally. The count of modules producing data streams is the number of modules (as defined by the taxonomy in the beginning of the section) that send one or more streams of data to another, distinct module. This count is collected as one indicator of the amount of data that is flowing through an experiment. The number of modules consuming data streams is obtained by counting the modules that accept one or more streams of data from another, distinct module during the course of the experiment.

The fifth step, selection of a regression statistical model, was not included because several of the necessary conditions were not achieved. One of the conditions not met was a random sampling of the global population since the samples largely came from pre-collected sources. Another set of conditions not met were knowledge of the global population size and variance of each of the dependent variables collected.

Table 1: Measures Used in Meta-Study

Measure	Measurement Range
Year Published	Integer: 1996 – 2015
Count Modules Producing Data Streams (*P*)	Integer: $P \geq 0$
Count Modules Consuming Data Streams (*C*)	Integer: $C \geq 0$
Count Total Number of Distinct Modules (*T*)	Integer: $T \geq 0,\ T \leq P + C$
Real-Time Inter-Module Communication	{Yes, No}
Data Standards Mention	{Yes, No}
Dynamism of Automation	{Static, Dynamic, None}
Adaptive Automation Exists	{Yes, No}

The total number of distinct modules is obtained by the number of individual modules according to the taxonomy. Since some modules may both produce and consume data streams, they are counted in both *P* and *C*, but only once in *T*. Thus, the total number of distinct modules may be less than the sum of consuming and producing modules. Real-Time Inter-Module communication is considered "Yes" if there exists at least one stream that is sent from a producing module to a separate, distinct consuming module which processes the stream and may or may not export the result. The most common example of this is a computational model predicting workload from one or more physiological data streams.

Data standards mentioned is coded as "Yes", when the authors mention details of the data infrastructure configuration. The threshold for coding a "Yes" is at least

mentioning what specific type of hardware is used for physiological or behavioral measures.

Dynamism of Automation has three levels: None, Static and Dynamic. “None” is coded when there is no automation that assists the human perform the task. “Static” is coded when there is computational assistance of the same task the human is performing, but the automation is configured to be on or off over the course of each entire trial (between experimental conditions). “Dynamic” is coded when there is computational assistance, and the type of automation changes during the course of a trial (within an experimental condition). The types of changes include varying the level of automation or triggering the automation on or off in real-time.

Each of the publications was reviewed for data architecture limitations that were either explicitly expressed by the authors or implied from the design. For example, it could be implied that an experiment had no ability to perform online data analysis across all physiological measures if each of the data streams were saved to unconnected storage devices.

Complexity Metric.

The core contribution of a data infrastructure software architecture is to manage the complex communication between data producing and consuming modules of an experiment. The reason for measuring complexity is to gather evidence for what type of design is required. A metric, rather than a narrative description, creates the ability to rapidly compare large numbers of infrastructure complexity for analysis. One approach to measuring the complexity of data communication within an experiment is to use features

of that experimental design to produce a complexity metric. In this section we will introduce a novel metric developed to quantify the complexity of data stream management. We first describe a very basic model with two features and build upon that to obtain a model that captures the complexity more accurately.

Two primary drivers are the number of modules within the experimental workflow that produce one or more data streams (*P*) and those modules that consume one or more data streams (*C*). A very basic metric to quantify the complexity of a data infrastructure is: $(P + C)$, where *P* is the count of data stream producing modules, *C* is the count of data stream consuming modules. *T* is the total number of distinct modules in the experiment. Since some modules may both produce and consume data streams they may be accounted for in both *P* & *C*, but only once in *T*. Thus: $P + C \geq T$

$$complexity = P + C$$

Requirements Gathering

This section of the chapter describes the activity used to review customer needs in order to specify the experiment data infrastructure requirements necessary to execute experimental designs. The primary goal of the activity is to synthesize software requirements that are common across human cognitive performance studies. This goal supports both research questions which seek to elicit a set of requirements and to investigate to what extent they are met by a central data mediator design pattern.

Another goal of the activity is to answer the question: What do researchers need from the software data management infrastructure in order to execute the experimental design? The answers to this question should be agnostic of specific instances of software

and hardware solutions. Requirements were drawn from the researchers because they are ultimately the end customer of the requirements elicitation.

Not all of the requirements engineering steps are used due to the scope of this thesis. We are not specifying a complete specification of a system that will be implemented in program code and are intentionally keeping aspects of the design vague to maintain broader applicability. The general steps for requirements engineering include: inception, elicitation, elaboration, negotiation, specification, validation and requirements management [24]. Negotiation and requirements management are two of the steps that are not being conducted. Negotiation is not conducted because this is an observational approach and the flow of information is only one way. Requirement management is not conducted because we are not pursuing the full development lifecycle through implementation in this research.

Inception

In the inception step, the customer and business need are defined. We defined both the stakeholder and user as the researcher who performs a human cognitive performance experiment. The users of the system may also include laboratory technicians who are executing experimental trials on the researcher's behalf. The business need is to answer a research question.

While answering a research questions consists of a large number of activities, the activity this research focused on is the collection, processing and storage of data during

the execution experiment. This activity occurs after the experimental design has been completed and before the analysis phase of answering the research question. Preparation of data for analysis is included, but not the analysis activities themselves.

Elicitation

Requirements elicitation consists of drawing out information from the customer. The two sources of data for requirements gathering were data infrastructure queries, as discussed by the previous section, and published literature on human cognitive performance experiments. The in-depth reviews were gathered from experiments that had the experiment's hardware and software equipment configuration defined.

Elaboration

The elaboration step begins to describe how the user, the researcher performing the experiment, will use the system. The scope of the use case was from start to finish of the actual execution of an experiment including data collection, data processing during the experiment (if applicable), and storage of the data in preparation for analysis. Since the experiment data infrastructure system only has one type of user, researcher, and one type of activity, experiment data management, a simple use case diagram was employed.

Specification

Specification adds detail to the elaboration step results. The specification was developed through a narrative use case. The purpose was to capture the data infrastructure system's behavior over the various sub-tasks of collecting, processing and storing the data during the execution of the experiment.

Validation

The validation step is used to examine the results for ambiguity, omission and inconsistency. The review of the requirements was conducted by answering three questions proposed by Pressman [24]. One limitation of the validation step was that it was only conducted by the author of this thesis and not by external entities such as a representative set of stakeholders. The questions used were as follows:

- Do any requirements conflict with each other?
- Is each requirement testable, once implemented?
- Is each requirement consistent with the overall objectives for the system?

Design Pattern Activities

The second research question asks how well a specific type of solution meets the requirements for executing the experimental design. This section describes both how the design was elicited from the software requirements, and also how that design will be evaluated against other basic alternative designs. In the results chapter, the central data mediator design pattern is compared against a data bus architecture and an ad-hoc setup. These two configurations are basic alternatives that highlight the benefits and drawbacks of the elements at the core of more complex designs. This section concludes with describing how the design will be evaluated.

The term chosen to describe the design, Central Data Mediator, implies several key design choices. The first of which is that there exists a distinct element that controls the streams of data, separate from any of the other existing modules in the experiment.

This element serves not only as a controller, but also provides a single interface for keeping modules from referencing each other directly. The design is primarily based on the Mediator Object behavioral pattern described by Gamma *et al* [25] and the Centralized Control architectural pattern described by Gomaa [26].

Design Pattern Specification

First is the software engineering process description of formalizing the requirements into a system specification and using standard design products to communicate the design. The architecture design choice was made between centralized, distributed, and hierarchical. Distributed was not chosen because it would have required manipulation of the mediator pattern which would add complexity without providing corresponding benefit to this problem domain. The complexity arises from having to create and maintain system state across multiple controllers. This may be beneficial when modules are spread across various networks, but in the case of most HCP studies, the experiment is a local operation. Hierarchical was not chosen because of the extra logic without benefit.

The degree of control that each module has over each other in HCP data infrastructures is very small. The case where this usually occurs is between the automation agent and the task environment, and is not significant enough to warrant the extra complexity. Centralized was chosen because it provides the ability for real-time synchronization without any of the unnecessary overhead of a distributed or hierarchical architecture.

The design description is purposefully kept void of discussion of any specific programming language. The pattern is intended for an object-oriented paradigm to leverage both information hiding behind a standard interface and classes of functionality. The abstraction of software details behind an interface makes the design more modular. The increased modularity enables the components to be modified or swapped with fewer changes to the software configuration.

The communication diagram in Figure 4 depicts a notional object-oriented class structure for the CDM. The top-left class, Colleague, is one of the modules in the experiment data infrastructure. The top-right class, Mediator, is the CDM module that serves as the hub. The Colleagues only need to be configured to connect to the Mediator regardless of internal changes to the Mediator class or other Colleague class instances that exist.

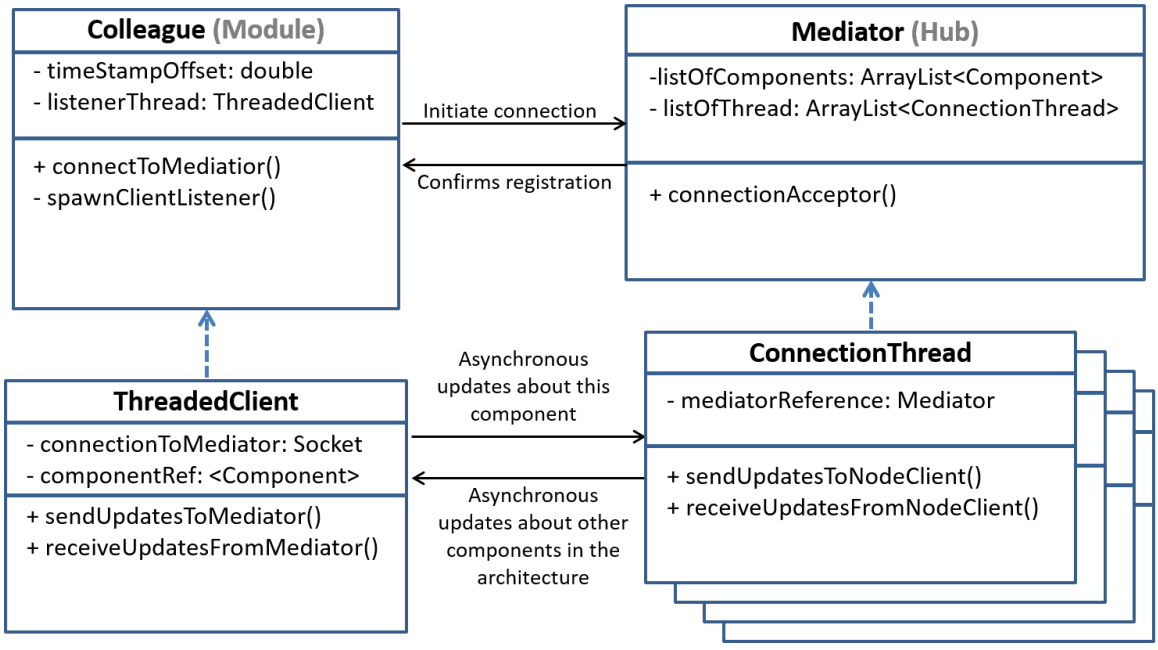

Figure 4: CDM Communication Diagram

Both the Colleague and Mediator classes have a separate subclass that serves to send and receive data message. This subclass runs in a distinct execution thread so that the module does not block execution while waiting for an asynchronous message of unknown arrival time. The Colleague class communicates once with the Mediator to establish a connection and so that the Mediator logs a new module to send and receive data from. All subsequent data messages are handled through the threaded subclasses and the control messages are passed between the parent classes.

Figure 5 depicts the general layout for the central data mediator within the context of an experimental architecture. The central component is the product of the proposed design. The logic for the design resides centrally in the CDM, however, software logic is also necessary for any modules that send data in both directions. Physiological sensors can stream data directly to the central hub using standard networking protocols such as TCP, UDP, or Bluetooth without having to implement any additional software on the sensor modules. Passive agents can also receive streams of data by only configuring standard network connection protocols for connection to the central hub.

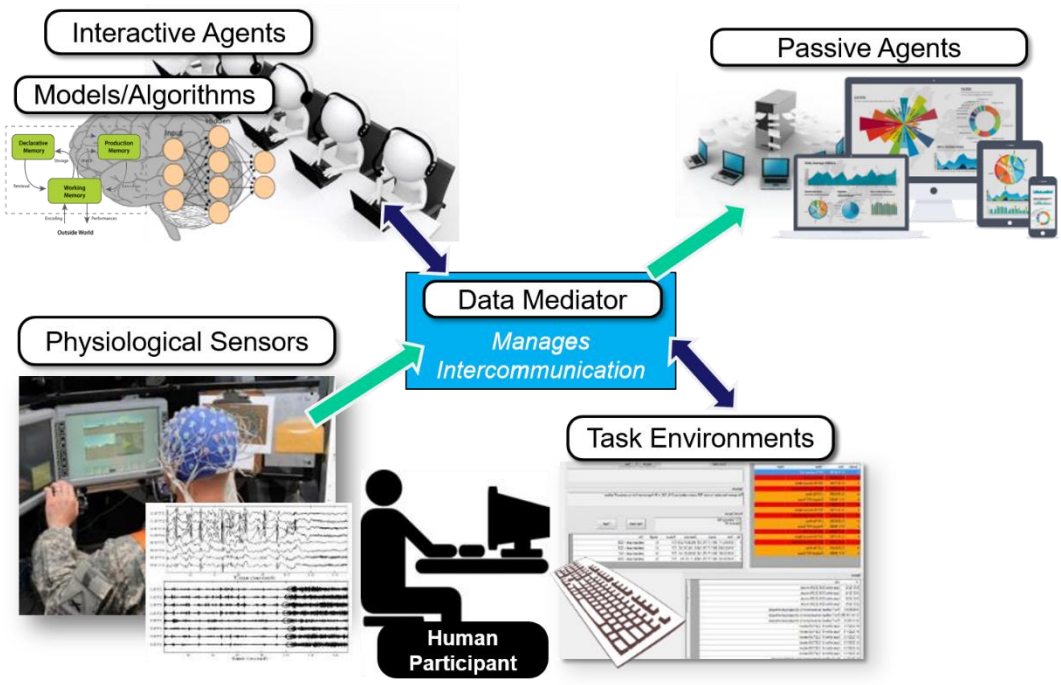

Figure 5: General Design – Central Data Mediator

Design Pattern Evaluation

This section describes the method for performing an evaluation of the CDM design pattern against two other possible architecture configurations, Ad Hoc and Data Bus. The designs are discussed in greater detail at the start of the evaluation in Chapter 4, but we provide brief introductions here for context. Ad Hoc indicates the lack of any formal design and an infrastructure where module communication is configured on a case-by-case basis. A Data Bus architecture specifies a shared communication medium is used by all of the modules use to broadcast produced data streams and listen for consumed data streams.

The format of the evaluation is a case study employing each of the three architectures, Ad Hoc, Data Bus & CDM. We perform the case study by examining the

effects of using the design at five phases of the experimental design execution where infrastructure design choices have an impact on system functionality and software development and maintenance costs. These phases are: 1) infrastructure design specification, 2) infrastructure construction, 3) running the experiment, 4) data extraction and analysis preparation, and 5) system reuse and maintenance.

After the effects of the three design during each phase are discussed, qualitative criteria are used to assess whether the benefits of using a CDM design outweighs the cost and disadvantages. Table 2 presents a notional effort using criteria to qualitatively assess whether a given data infrastructure warrants the use of the CDM design pattern.

Table 2: CDM Evaluation Rubric

ID	Criteria	Red (CDM Not Warranted)	Yellow (CDM Use Uncertain)	Green (CDM Warranted)
1	Are real-time module interactions necessary?	No	N/A	Yes
2	Will this data infrastructure be used for HCP studies in the future?	No	Unsure	Yes
3	What percentage of physiological sensors can export their data in real-time [RT] (versus limited to local, on-device storage)?	$RT\ Sensors \leq 50\%$	$50 < RT\ < 75$	$RT\ Sensors \geq 75\%$
4	How many distinct data streams must be aligned for the analysis phase?	$Streams \leq 3$	$4 \leq Streams \leq 6$	$Streams \geq 7$
5	Total module count, *T* (P+C)	$T \leq 5$	$6 \leq T \leq 10$	$T \geq 11$
6	How often are the modules modified (over the course of a particular study)?	Never	Once/Unknown	Twice or more
7	Are there an unknown number of modules that may need to be configured dynamically during the execution of a trial?	No	N/A	Yes

The method for assigning a final assessment from the rubric is based on the count of criteria that register for each color. If more criteria are coded green than red, then using the CDM design is warranted. Otherwise, if there are less criteria coded as green compared to red or there is a tie, then a CDM is not warranted.

Chapter Summary

This chapter built on the introduction to current human cognitive performance studies and the activities being conducted to improve data infrastructure design and capabilities. This chapter presented a method for measuring complexity and capturing requirements, with the goal of helping to determine the necessity of applying software architecture design to workflow data architectures. Presented next was a method describing how to conduct an analysis of a specific software architecture design. The design chosen was based on attempting to meet the identified current and future requirements of researchers in this field. The next chapter discusses the results of conducting these described methods.

IV. Results and Discussion

This chapter discusses the results of each of the research activities maintaining the same order as their methodology was presented in the last chapter. In the opening of this chapter, we briefly summarize our research goals to provide context for the presented results. Second, we review the detailed software and hardware descriptions from SME interviews and Experimental Design reviews otherwise not available in published literature. Next, the coded results from the meta-study are examined for data infrastructure complexity and real-time usage trends across the samples. Then the list from the requirements gathering activity is discussed to establish a basis across the observed samples. Finally, an evaluation of the Central Design Mediator design pattern is presented to exhibit the effects of the design on the resulting data infrastructure.

Data Infrastructure Queries Results

This section discusses findings gleaned reviewing hardware and software configurations of human cognitive performance experiments. The reviewed sources contained equipment configuration details that were not included in published literature. Even though the sample size (n = 7) was very small, the findings can be extended to a larger range of experiments because the number and types of modules as well as the research goals were very similar to those in the meta-study. The data from the interviews and reviews is located in Appendix B: Data Infrastructure Query Results.

Overall, we observed low rates of interaction between modules during the execution of experiment trials. One common example of interaction was the output of

physiological or behavior data streams to a computational module that produced a workload level. Another, less common interaction was between the automation controller and the task environment. Additionally, the modules that did interact had a low level of interdependence, the need to share their exact current state. One reason for this was that the majority of message passing was one-way.

In general, the execution of each module did not depend on the exact state or timing of others. While data streams were crucial as inputs from one module to another, knowledge of the exact state of another module was not necessary. The message complexity measurement did not provide much value because the experiments were either conducted with all the components on the same physical machine or local area network (LAN). The quantity of data flowing during the experiments was insignificant compared to the capacities of the networks on which they resided.

Meta-Study Results

This section includes the results of the meta-study of published human cognitive performance experiments. The purpose of the meta-study was to determine the requirements of human cognitive performance experiments being conducted throughout the research community. Therefore, the focus while reviewing each published experiment was the data architecture content and configuration.

There were several goals for analyzing the meta-study results. The first was to identify whether a trend of increasing complexity of the experimental workflow exists. A second was to capture common factors of the designs used to build each experiment's data infrastructure, such as support for real-time communication or dynamic levels of

automation in the task. Determining the existence of any discussion on designing the data infrastructure to support executing the experiment with the literature was a third goal. The list of studies included in this meta-study is located in Appendix C: Meta-Study.

The published research on human cognitive performance experiments was chosen based the minimum qualifications of the scope discussed in the last chapter. To recall, the included studies consisted of human cognitive performance experiments in which a human subject engages in a task and from whom behavioral or physiological data streams are collected and then processed or stored. The sampling of the publications was ad-hoc; not entirely random, nor systematic. The publication years of samples ranged from 1996 to 2015. The distribution of experiments that were collected over that range of years is presented in Figure 6. The number of published experiments collected for each year does not necessarily represent the proportion of experiments published in the corresponding year.

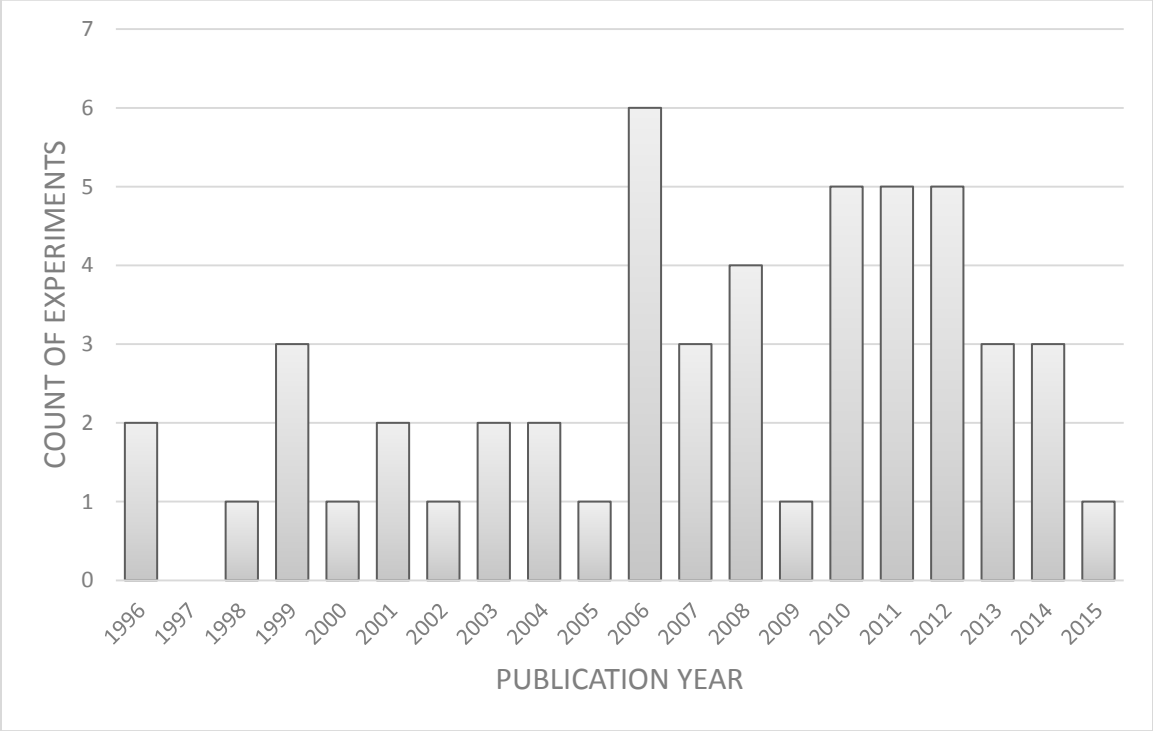

Figure 6: Distribution of Surveyed Experiments (*n=51*)

While all of the published experiments collected fall under the umbrella of human cognitive experiments, there was a focus on gathering experiments for which the proposed Central Data Mediator is most applicable. These research focus topics include Adaptive Automation (AA) and the research efforts to support AA. The supporting efforts in general consist of understanding how humans react to various levels of automation (LOA), automation invocation methods and determining operator functional state (OFS). The key properties of these topics are real-time data stream management and tasks that dynamically change throughout a trial.

Meta-Study Results.

This section reviews the results of the meta-study first to investigate for the conjectured trend of increasing complexity in data infrastructures. Identified trends would improve our understanding of how the data infrastructure requirements may change in the near future. One of the most basic methods to assess complexity is to quantify the number of distinct modules producing or consuming data streams.

Figure 7, shows the average number of modules for each experiment over the time range which the meta-study covered. The producing and consuming modules are counted according to the taxonomy from Chapter 3. It is important to keep in mind that the total number of modules measure, T, in an experiment will be less than or equal to the sum of modules producing data streams and the number of modules consuming data streams ($P + C$) because some modules may both produce and consume data. Instead of representing the sum of all the data streams, the total number of modules represents the number of distinct modules that appear in an experiment.

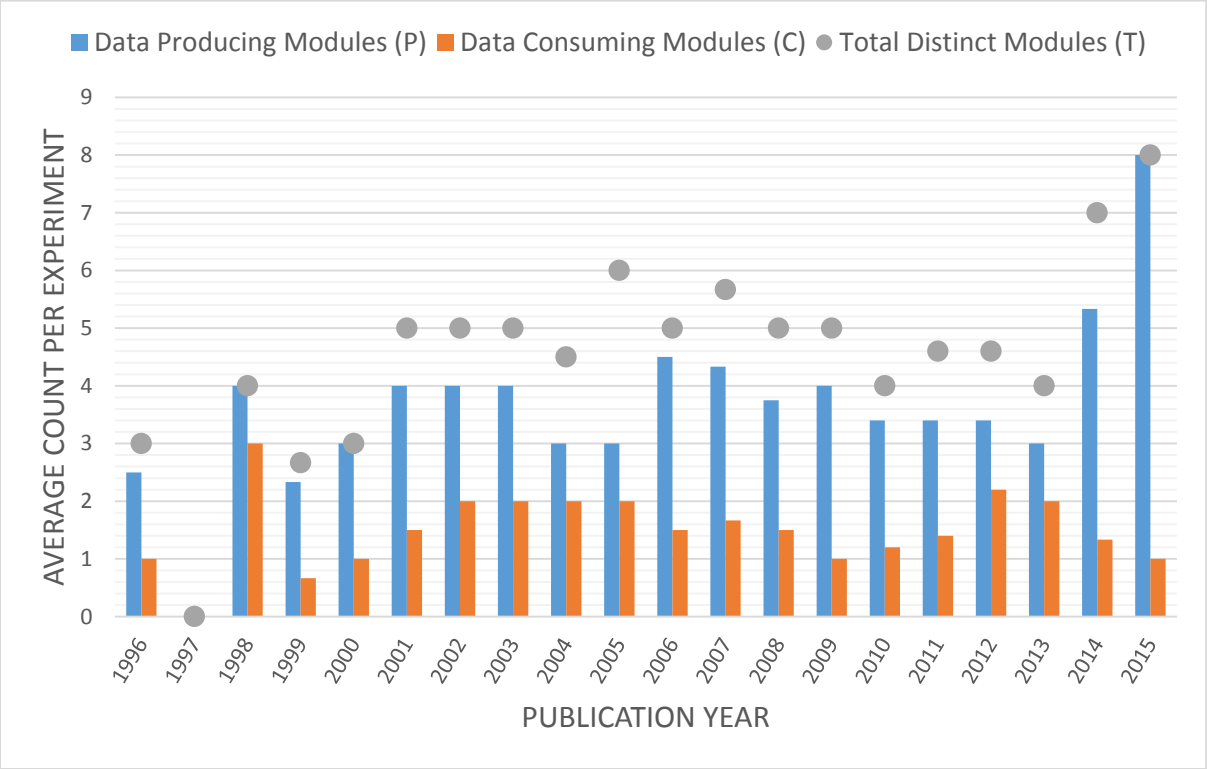

Figure 7: Average Modules per Experiment over Time

One of the most obvious observations is that there are always more producers of data than consumers. This indicates that the data streams are consolidated and sent to one module or their stream isn't sent anywhere at all and is instead stored locally. The graph also shows that the metrics remain fairly stable over time. Since the number of components that data infrastructures are designed to support remains stable over time, this may indicate that the software requirements for the infrastructures may also remain stable.

One limitation of this graph is that it oversimplifies the measure of complexity that exists within the data management infrastructure for each experiment. A simple

count of the modules does not capture the complexity of that single data stream, such as whether it is a subjective workload captured once per trial or a set of EEG features sampled at 2.0 MHz. Another aspect of complexity that is not captured is the real-time data streaming requirement versus saving individual streams to a persistent storage.

The following equation was used to develop the plot in Figure 8 that provides a distribution of all the complexity values assigned to the observed experiments. Each of the points is the complexity value for an individual experiment.

$$complexity = (P + C)$$

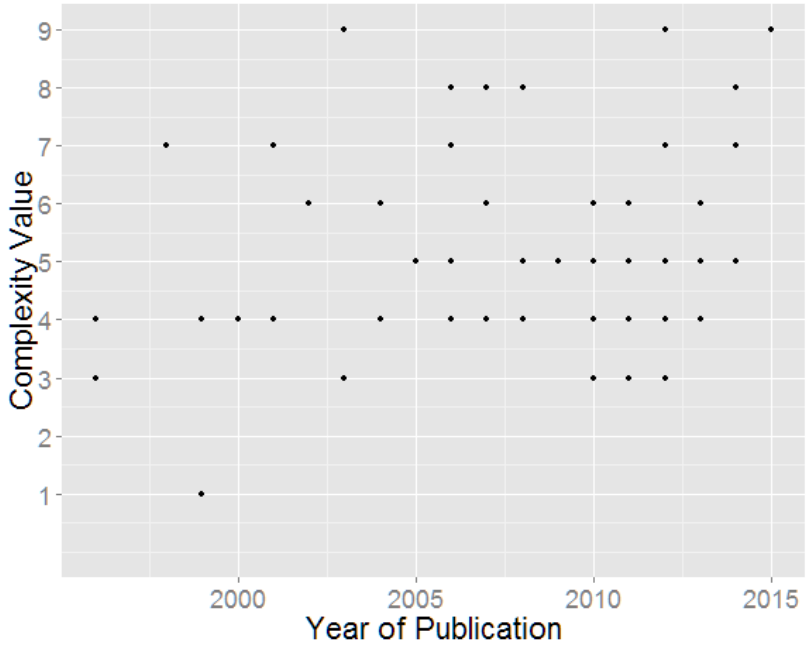

Figure 8: Trend of Complexity over Time

In Figure 8, the distribution over the years is the same as the original histogram in Figure 6. The distribution of complexity values does not show a definitive increasing trend when applied over all of the observed experiments. Next we observed whether there

was a subset of the experiments that demonstrated an increasing trend. Whether the experiment included any form of automation was chosen as the feature to create distinct graphs for Figure 9.

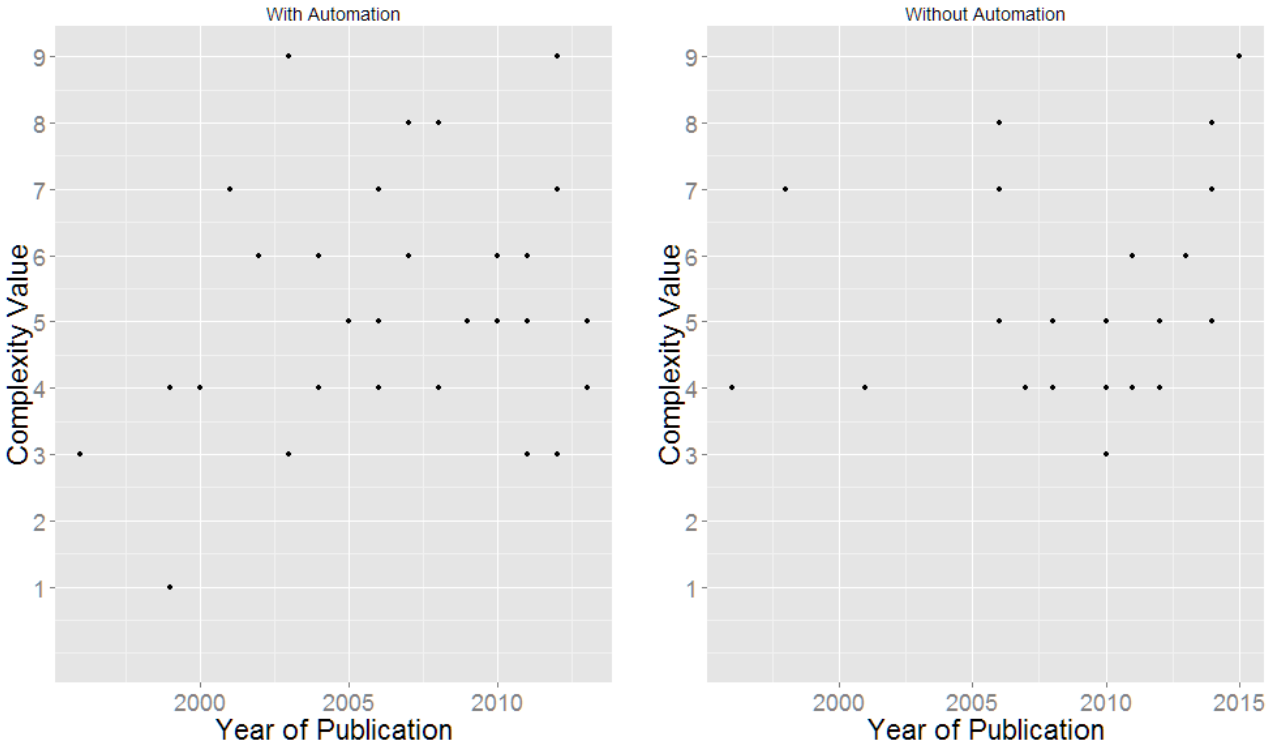

Figure 9: Complexity Values With and Without Automation

In Figure 10, a box-and-whisker diagram of the same complexity values is used to better visualize the median values over time. The observations are grouped by every 4 years in order have enough observations per box to be meaningful. This complexity value distribution does not show a definitive increasing trend, but it does not rule out the existence of a trend even for this simple count of modules.

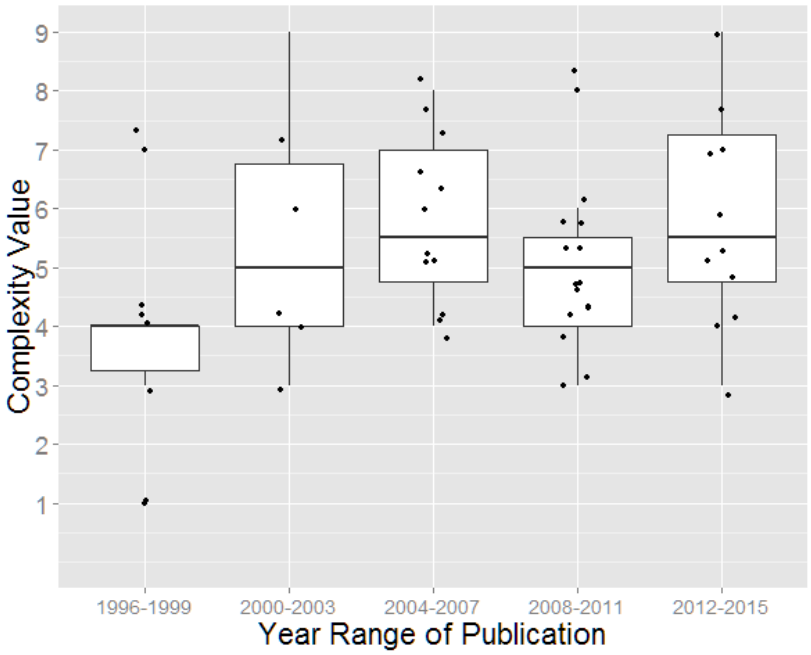

Figure 10: Complexity Metric Grouped by Year Range

Task environments become more dynamic as real-time modification by automation and augmentation controllers is being added to the experiment systems. Dynamic task environments may mean an increase in complexity of the software infrastructure because the state of the task can no longer be known a priori or implicitly from the conditions of a trial. Instead, a time series of system or task environment states must be recorded. This task state time series must also have the ability of being aligned to each of the other data streams so that physiological and performance responses can be mapped to the subject's actions and task environment current state.

The complexity metric does not currently take into account the complexity of the data stream contents. For example, subjective workload collected once per trial requires

much less complexity than EEG channels sampled at rates as high as 2.048 MHz over multiple frequency bands.

Meta-Study Limitations

The findings from the sum of the studies presented in this chapter are limited to explaining only the samples within the observational study. While the publications used in the meta-study were not systematically chosen within the constraints of the scope, they were also not randomly sampled from a global population. The distribution of the samples of the complexity feature, such as the complexity score or level of adaptive automation dynamism, cannot be shown to be statistically representative of the whole population of human cognitive performance experiments. For this reason, the samples cannot be used to provide arguments for the entire population of human cognitive performance experiments or even real-time adaptive automation studies.

Meta-Study Summary.

This section provided data gathered from the human cognitive research community to examine the current state and trends of complexity with respect to the data management infrastructures. The lack of computational models that can accurately assess operator functional state in real-time may be one reason that a trend towards more complexity may not exist. Without existing well-performing models, research is focused more on collecting data to develop the models rather than studying the effects of using them in real-time, such as with adaptive automation. The next section details the specific requirements for data architectures that make up the complexity of experiments discussed in this chapter.

Requirements Gathering Results

This section presents the results of the requirements gathering process as defined in Chapter 3. The requirements gathering process is conducted with human cognitive performance researchers as the customers. Inception and Elicitation are not included in the results. There is no new content for Inception and the Elicitation results were covered by the Data Infrastructure Queries Results.

Elaboration

We provide a very simple stick figure use case to demonstrate the primary user and activity.

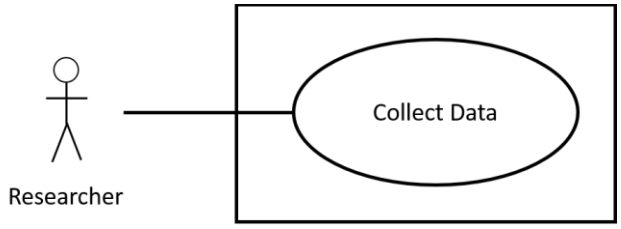

Figure 11: Basic HCP Experiment Use Case

Specification

The specification consisted of drawing observations from the SME discussions to build the following narrative use case.

Use case name: Conduct human cognitive performance experiment

Summary: The human subject conducts a task and physiological and behavioral measures are collected in order to answer a research question.

Actors: Research Scientist

Preconditions: The hardware and software to run each module is complete. The physiological sensors are able to output their measurements in real-time.

Main sequence:

1. Research scientist configures all modules producing (e.g. physiological sensors) and consuming (e.g. OFS model) data streams to begin to produce and consume data streams respectively.
2. The research scientist begins the task and experimental trial.
3. The data infrastructure routes data between modules during the experiment execution. Some modules store the input through the end of the experiment. Others process input data streams and output a result as another data stream.
4. The research scientist collects the data streams into a single logical (e.g. hard drive or network storage) location and time-aligns data in preparation for analysis.

Alternative sequences:

Step 4: Some of the data from the experiment was not sent to a shared storage location. Instead the data was stored locally at the producing module. Locally stored data from multiple distinct locations needs to be combined and time-aligned.

Postcondition: Data has been collected in preparation for filtering and analysis post experiment completion.

Following is a bulleted list of requirements gathered from completed and planned human cognitive performance studies. The sources for the use cases were the meta-study, SME discussions and Experimental Design reviews. The ordering does not suggest any prioritization.

Software requirements:

1. Data streams from disparate processes not operating on the same system clock shall be time-series aligned.
2. Modules within the experiment shall have the ability to receive multiple streams of data from one or more other modules in real-time.

3. Modules within the experiment shall have the ability to send multiple streams of data to one or more other modules in real-time.
4. Multiple data types shall be handled simultaneously. Examples are double floating points, integers, strings, binary, images and video.
5. The available physiological hardware sample rates shall be maintained.

Validation

The validation step answered the following three questions:

- Do any requirements conflict with each other?

None of the requirements are mutually exclusive. Neither does meeting any one of the requirements hamper the ability to meet any of the others.

- Is each requirement testable, once implemented?

The first requirement creates a specific end result that can be tested. One method for testing alignment would be to use an existing, correctly-aligned data set to compare against the results of the data infrastructure after replaying the distinct data streams. The second and third requirements represent functionality that can be tested, such as by providing input and measuring the time until the corresponding output. The fourth requirement can be verified by providing the system with the necessary file types and observing that each are collected, stored or processed correctly. The fifth requirement can be verified by noting the sampling rate at the source and confirming that rate where the data is ultimately stored.

- Is each requirement consistent with the overall objectives for the system?

The first three requirements specify aspects of real-time data management functionality. Real-time data management is necessary for several types of key human cognitive performance research such as adaptive automation that need to assess OFS in real-time. The fourth requirement is supported by the need to manage many data formats used in HCP experiments. In addition to the many formats of physiological and behavioral measures, HCP experiments also store video recording as a data source for analysis, such as in Recarte *et al*'s work [27].

Design Pattern Evaluation Results

This section discusses the effects of using a Central Data Mediator architecture design on experimental workflow design and implementation. In order to make the discussion more salient, we present a case study using an experimental design from the literature. Real-time support is a minimum requirement for the architecture because it is a necessity in a subset of the experiments. To support a population-wide application, the architecture design then must also support real-time systems.

This section examines the case study application of the CDM design pattern to a published experiment in comparison to both an ad hoc and data bus configuration. To review from Chapter 3, we perform the case study by examining the effects of using the design at five phases of the experimental design execution. These phases are: 1) infrastructure design specification, 2) infrastructure construction, 3) running the experiment, 4) data extraction and analysis preparation, and 5) system reuse and maintenance. Before the application is discussed, the experimental design of the

publication is introduced. Then we list the experiment's data infrastructure modules. Each of the infrastructure designs discussed in the evaluation are introduced next. Finally, we discuss the consequences of the CDM design pattern in each phase.

Case Study 1

The experiment used for the case study was originally conducted in 2012 by Dorneich *et* al [28]. The research goal of the experiment was to measure the costs and benefits of an adaptive automation task interruption manager. The task, conducted in a physical environment, required the subjects to perform concurrent navigation and target spotting tasks while receiving informational and instructional messages. The adaptive automation module, Communications Scheduler, modified how messages were presented to the subject based on a workload assessment and task context. The workload assessment was provided in real-time by a cognitive state classifier using EEG and ECG (heart rate) data streams.

There are several other modules in the experiment in addition to the Communications Scheduler and the cognitive state assessor, all of which appear in each of the diagrams. The physiological stream producing modules were EEG and ECG sensors. The task environment contains several components that are the internal generators or recipients of data streams within the module. In order to examine the design more fully, two components are added. The data storage and distraction model were added based on the authors conclusions that a deeper understanding of the human's state and nonverbal cues leads to improved human-machine interactions. The first of the additions is a distraction model that uses eye tracking information. The second is storage

of the each of the measures and model output generated throughout the experiment to enable post-trial analysis. The original setup collected several task performance metrics.

The first configuration, shown in Figure 12, does not consists of an overt design for connecting each of the components. Each of the connections are independent of each other, and a connection is configured only when necessary between two modules. The connection must be configured for both the producing and consuming modules.

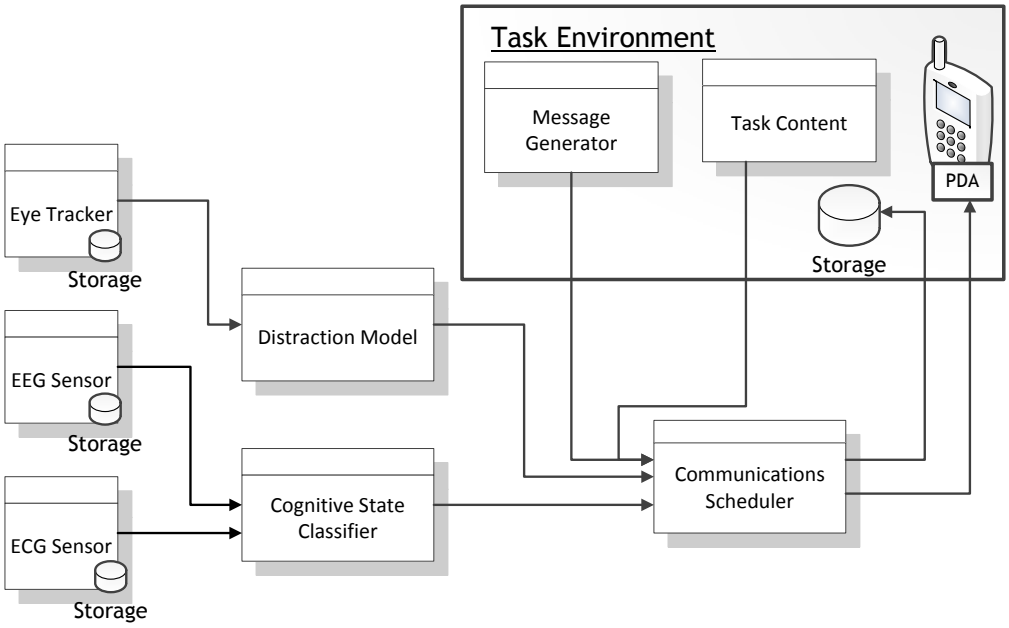

Figure 12: Ad Hoc Module Concurrent Communication Diagram

The next configuration, shown in Figure 13, provides a single interface for each module to communicate with others. The bus is a message passing medium, but it does not contain any logic that processes the data. There may be extra logic necessary to filter

out the messages not destined for a specific module because each message is broadcast to all of the modules. This logic is located in each of the individual modules.

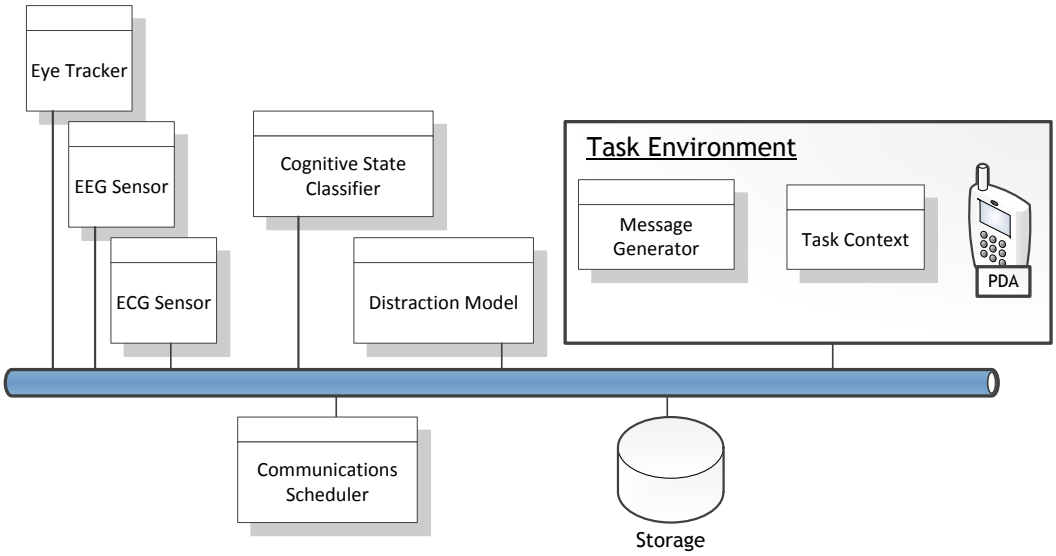

Figure 13: Data Bus Module Concurrent Communication Diagram

The architecture of the Central Data Mediator (CDM) is depicted in Figure 14. In this diagram, the double sided arrows signify that data may pass in both directions. Flow in data in both directions is a capability for every connection, but physiological sensors only produce data and so the arrows for those modules are drawn in one direction. The CDM is considered an additional module that exists in software, but does not inherently require additional hardware to implement.

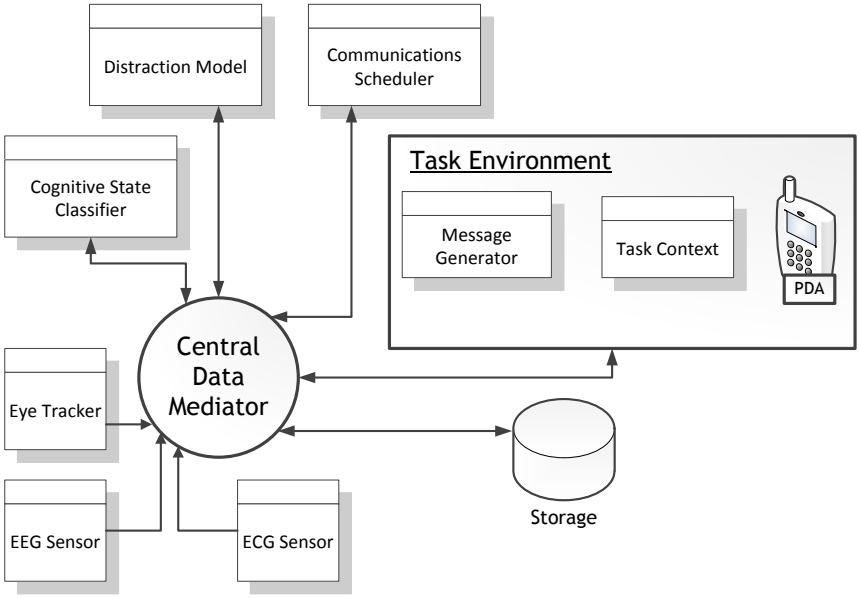

Figure 14: Central Data Mediator Module Concurrent Communication Diagram

We now discuss the effects of the three designs with respect to each of the five phases.

Infrastructure Design Specification

In the initial infrastructure design specification, requirements gathering and results will not differ because they are driven by the same research goals. The process of creating the specification for how the connections are configured is more involved for the data bus and CDM configurations than for the ad hoc setup. The extra work is due to establishing or adopting a single interface that all of the modules will use for communication external to themselves. This interface must support the functionality of all of the modules in the data infrastructure. The interface will have more complexity in the CDM configuration compared to the data bus because data destination information must also be included.

In the design specification phase, the CDM has the highest relative cost, but the cost is mitigated to a degree by a high potential for reuse even if the modules are completely different. The reuse statement is also true for the data bus. The ad hoc setup will still require implementation of logic for each connection to function, but an established design is not necessary.

Infrastructure Construction

One of the key differences with respect to the software implementation during the infrastructure construction phase is amount of software logic that must be implemented for each individual module. For each module in the ad hoc configuration, there are multiple types of interfaces that need to be configured. These include physiological data streams to models, models to the task environment, and task environment updates to automation controllers. There is configuration that needs to occur at each module for the data bus and CDM designs, however, the configuration is only a slight modification for each.

Running the Experiment

While the experiment is running, the CDM natively offers the ability to process all of the data in real-time because it has access to all of the streams in a single location. Real-time alignment is achievable with the ad hoc and data bus architectures, but additional configuration or equipment is necessary. In the ad hoc setup, the modules can work off of a shared start time, however, this approach is prone to time drifts inherent in different hardware. An approach for the data bus architecture is to include additional equipment to run Network Time Protocol (NTP) which can broadcast the current time to

all of the modules to prevent time drift. Both of these methods require the implementation of additional software logic.

Data Extraction and Analysis Preparation

This phase consists of gathering the data produced and collected during the course of the experiment. The data is moved to a location and converted, if necessary, to a format such that post-completion analysis can be conducted. The activity we focus on for this phase is moving and time-aligning the data because of the large amount of resources it takes to accomplish.

Using an ad hoc configuration, there are two general solutions for combining and time-aligning the collected data into a single location. One method would be to store data streams at the module from which it is produced. An alternative is to configure the modules to send all of the data streams to a central storage location. Both of these options require significantly more configuration to achieve the same result as compared to the data bus and CDM designs.

In the data bus design, all of the data is already being broadcast and once a storage module is connected, it can collect every data stream. The storage module must contain logic to align or assign timestamps if they do not already exist for each data point in a data stream. Similarly, with the CDM design, adding a central storage module only requires a single connection to the CDM module. The difference from the data bus design, however, is that the logic for aligning and timestamping the data points resides within the CDM module. The central location of logic in the CDM is advantageous when

the storage module changes and the logic must be reconfigured or reimplemented in the storage module in the data bus design.

System Reuse and Maintenance

Software maintenance refers to updates to the functionality or format of the modules. For example, this may consist of an entirely new version of a computational model or more simply an update to the parameters that the model outputs. The CDM and data bus configurations benefit the reuse phase because of their modular standard interface. This allows reuse of modules from previous experiments in a "plug and play" fashion. The effect is a reduction in the amount of source code that needs to be configured. This applies to both the central mediator and the module itself. In the ad hoc configuration, if the modification or replacement of a module affects the communication format, all of the other modules it is connected to must also be modified.

Case Study 1 Summary

The results of the criteria assessed for the experiment in Case Study 1 are in Table 3 below.

Table 3: Case Study 1 Evaluation Rubric

ID	Criteria	Red (CDM Not Warranted)	Yellow (CDM Use Uncertain)	Green (CDM Warranted)
1	Are real-time module interactions necessary?			Yes
2	Will this data infrastructure be used for HCP studies in the future?		Unsure	
3	What percentage of physiological sensors can export their data in real-time [RT] (versus limited to local, on-device storage)?			$RT\ Sensors \geq 75\%$
4	How many distinct data streams must be aligned for the analysis phase?		[Streams = 6] $4 \leq Streams \leq 6$	
5	Total module count, *T* (P+C)		[T=6] $6 \leq T \leq 10$	
6	How often are the modules modified (over the course of a particular study)?	Never		
7	Are there an unknown number of modules that may need to be configured dynamically during the execution of a trial?	No		

Since the total number of criteria coded green and red are tied, this experiment would not warrant the use of a CDM design without additional extenuating factors.

Case Study 2

The second case study is performed on the experimental design as described by the primary investigator of the 4th SME interview. The data infrastructure was described by its developers as a type of universal data bus or a data bus that contained logic for managing and formatting data. A visual diagram of the configuration is provided in Figure 15. There is a single interface for all new modules to connect to. There are four physiological modules producing at least one data stream. For behavioral and task situational data, there are three data producing modules, to include the status of the

human-computer interface (H-C Interface) presented to the operator to inform the task augmentation controller.

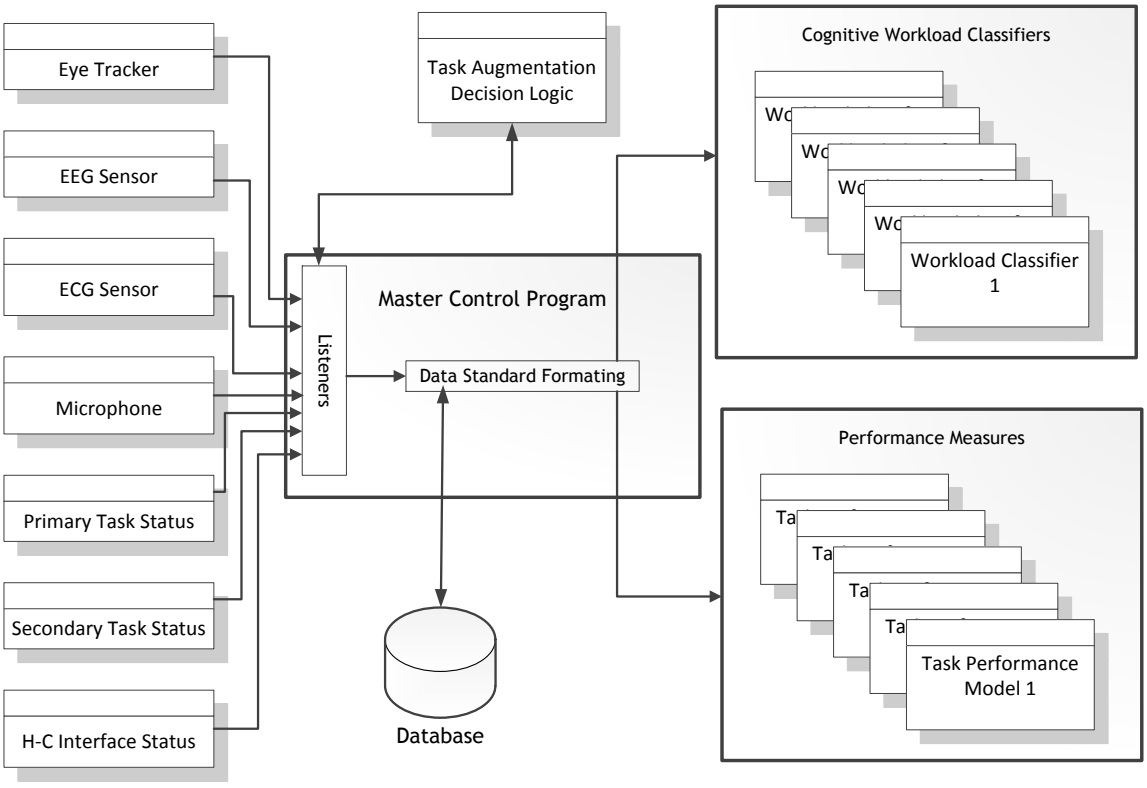

Figure 15: Concurrent Communication Diagram – SME Interview 4

Rather than using a single performance or cognitive workload measure, the experimental design creates multiple of each. Multiple models capture various aspects of either performance or workload using different sets of inputs. The results are consolidated into a single continuous measure. All of the data is also stored in a central database.

The results of the criteria evaluation are in Table 4 below.

Table 4: Case Study 2 Evaluation Rubric

ID	Criteria	Red (CDM Not Warranted)	Yellow (CDM Use Uncertain)	Green (CDM Warranted)
1	Are real-time module interactions necessary?			Yes
2	Will this data infrastructure be used for HCP studies in the future?			Yes
3	What percentage of physiological sensors can export their data in real-time [RT] (versus limited to local, on-device storage)?			$RT\ Sensors \geq 75\%$
4	How many distinct data streams must be aligned for the analysis phase?			$Streams \geq 6$
5	Total module count, *T* (P+C)			$T \geq 11$
6	How often are the modules modified (over the course of a particular study)?		Once/Unknown	
7	Are there an unknown number of modules that may need to be configured dynamically during the execution of a trial?	No		

With the count of criteria coded green, the second case study experiment warrants the use of a CDM. This does not mean that a CDM is the only solution for the data infrastructure, but that it would be an appropriate solution. A greater count of criteria coded as green indicates that the benefits of using a CDM design would outweigh the costs.

CDM Analysis Summary.

The example applications of CDM to various experimental designs show that the benefits and drawbacks are dependent on the makeup of the experiment. There are some types of experimental designs that the CDM is especially suited towards. These include

those experiments with many distinct sources of continuous data flow that must be processed in real-time during the execution of an experimental trial. There is also a time cost that applies to adapting the laboratory's processes to building data infrastructures to a new standard. Additionally, the time for the support personnel building the data infrastructure to learn this new paradigm may be another drawback that must be balanced by the time-alignment and modularity benefits.

Discussion

While the research goals of human cognitive performance experiments vary, there are similarities in the way data is collected and managed during the execution of an experiment. Leveraging the requirement commonalities, specifically software design patterns, provides several benefits. A community-driven set of solutions can lead to additional capabilities and reduction of the resources necessary to construct, execute, and analyze the experiments.

The results show that many experiment data infrastructure requirements are common across the majority of these types of experiments conducted. It was also argued that the complexity of the data communication will only continue to increase. There are still benefits to a continued effort into software design in this domain even if the complexity at this current time does not demand the use of specialized software architectures. Benefits include reduction in resources necessary to build new or modify existing experiment data infrastructures and native support for time series alignment of data streams from modules with distinct system clocks in real-time. As research in the near future increasingly employs adaptive automation requiring the management of

multiple real-time streaming data modules, data infrastructures will need to grow more complex to support those requirements.

V. Conclusions and Recommendations

Summary

Software design requirements were gathered from human cognitive performance experiments and used to develop and evaluate a design pattern. The software requirements, design pattern specification and evaluation provided insights into the data infrastructure necessary to achieve the various research goals of human cognitive performance experiments.

The advantages of a software design pattern are realized as the complexity of data management increases. The primary sources of this complexity in human cognitive performance experiments are the number of distinct, separate process streams of high fidelity data and real-time synchronization requirements. The more the two sources exist in an experiment, the greater the necessity for a design to manage the complexity is needed.

Even though we did not show a trend in increasing or decreasing complexity, the assertions of authors from human cognitive performance literature suggest an increasing trend. While the authors may not address the topic of complexity in their software data infrastructures explicitly, they argue for future work that implies an increase in complexity.

One of the critical components required for adaptive automation is the operator assessment or state determination function argue Barnes, Parasuraman and Cosenzo [29]. As early as 2001, Russell and Wilson argued that in addition to physiological data,

performance and situational data could result in higher accuracy of operator state [30]. This would result in the need for additional streams of data to be aligned and further add to the complexity of an experiment's data infrastructure.

In order for the machine within the human-machine team to provide the appropriate type and amount of aiding at the optimal point in time, the machine must have a sense of the human's state. There are many methods for determining the components of the operator's state in order to make the best decisions. The methods primarily consist of observing the operator's behavioral and physiological metrics. However, it may be beneficial to include as many operator functional state assessments as possible from various features argue Durkee *et al* [31]. Not only may more assessments be better, Fairclough and Venables argue more complex interactions of physiological features may offer further benefit [32]. Therefore, the complexity of experiments will continue to increase as additional features are collected and streamed to operator functional state assessment models and the output of those models are streamed throughout the experiment to inform changes in the task or other components of the study.

The following are some practical applications for the results of the research. The design pattern could serve as a starting point for future, actual instantiations of software data infrastructures. Improvements on the design could be captured and published to continue evolving the design to best meet the needs of the human cognitive performance community.

Development and improvement of data infrastructure suffers as a result of the lack of discussion within the publication methodologies. There are likely several reasons that a discussion of the specific software configuration for managing data within the experiment is not included. One of the reasons is that there is little direct research contribution for discussing the design of the data infrastructure or the challenges. Another is that there are no human cognitive performance domain standards for building one. Without common design standards to reference, it requires a large portion of an article to describe the software infrastructure. The majority of the time, the authors consider the portion too great and do not include any discussion. If general architecture software designs for this domain existed in literature, a single sentence describing the type of design used may be sufficient and beneficial.

Benefits of using a centralized design were identified, but did not come without cost. The benefits included the temporal alignment of data streams in real-time, a single interface for programming communications, and a reduction in reuse costs when compared to an ad-hoc configuration. The costs for these benefits consist of a higher upfront cost of software planning and implementation. The benefits are realized more so as the quantity of data streams increase and the complexity of interaction between the modules of the experiment increases.

While an analysis of the existing state of complexity suggests that a Central Data Mediator design is not essential to meet the currently standing experimental design requirements, it also shows promise for positive impact in the present and near future. One use case is to address the trend of increasing complexity of experimental design

requirements for which a CDM or similar design may be necessary to meet in the near future. The second benefit is that the design still offers a reduction in the resources necessary to design and implement the data infrastructure.

The reduction of resources comes from the modularity of data stream producers and consumers when implementing a mediator design pattern. Another source of the reduction of resources is the ability to handle time-series alignment in real-time. This provides additional capability to the researcher and reduces work necessary after the completion of the experiment.

Contributions

This thesis provided an example of a software design pattern mapped to a novel domain, the data infrastructures of human cognitive performance experiments. The application of domain knowledge was accomplished by collecting observations from both published and unpublished sources. This activity could serve as lessons learned when repeated with a greater sample size or a different design paradigm.

This research also produced a preliminary approach to capturing and analyzing the software complexity in the human cognitive performance domain. A basic metric was presented and the limitations were discussed. This could be used as a basis for future work to generate more accurate models of complexity.

Future Work

A partial list of metrics for human cognitive performance data workflows and software architectures was discussed in this work. There are several potential benefits for future work to gather additional metrics and compare the requirements analysis against

the results of the presented method. One benefit is further validation of the findings from this research. Another benefit is a deeper understanding of the root causes of complexity in the experiment data infrastructures and how to address them.

To further the concept of providing a common design architecture that can be used as a template, implemented software solutions could be added to the available resources along with the design. The goal is to build a library of code that follows the design, but is still highly configurable. The flexibility needs to remain in order to meet the custom requirements of the vast array of research goals and subsequent experiment designs. An additional desired quality would be implementations and documentation such that the resources could be applied by someone who is not a software developer.

The ultimate goal is to produce resources that reduce the burdens of constructing, using and maintaining software data infrastructures in human cognitive performance experiments. Toward that goal, this thesis provides groundwork for further research and discussion. With contributions from the academic and operational communities, openly available code libraries and enhanced, validated designs may lower the barriers to entry for new research groups and enable established labs to take their research even further.

Appendix A: Data Infrastructure Query Contents

The purpose of this document is to gather metrics on the portions of the human cognitive performance study that produce, consume and route the data during the course of an experimental trial. We desire to understand HOW you move and store your data more than WHY you are collecting it. Additionally, we are more interested in the format rather than the specific content (factors and levels) of your data. For example, the scope includes physiological data streams, how that data gets to a cognitive workload model, and where the output of the model needs to go (such as storage, visualization, or a level of automation controller). The scope does not include the construction or analysis of the model, an analysis of what physiological measures are used, or the specific results of the trial.

In order to clarify the questions included and the desired format of the answer, brief descriptions of the measure, examples of the subject in question in a hypothetical experiment and examples of how it could be answered.

Scope of studies for this research

1. The studies reviewed will be Human Cognitive Performance experiments
2. The studies will have at a minimum:
    a. A task environment
    b. One human participant and/or one or more machine agents that can take actions in the task environment
    c. Post-study storage and analysis of collected data
    d. Real-time or replayed psychophysiological measures, at least one instance (i.e. heart rate, EEG, pupilometry, etc)
3. The studies may also have:
    a. Computational elements that do not produce actions in the task environment, such as:
        i. Workload prediction/assessment algorithms
        ii. Cognitive architecture to produce workload estimation
        iii. Data visualization logic/products

Background Questions

1. Provide a publication abstract level description of the study
2. Describe the data that each trial collects, such as:
    a. Human subject action and performance metrics
    b. Individual streams of physiological data
    c. Computational model outputs
3. Describe the process of handling data from its production source to where it remains after the experiment trial is complete. For example:
    a. The human participant's mouse and keyboard actions are generated in the task environment and are sent to a computational model where performance is assessed.

Measures to collect and Questions to elicit those metrics

1. Standard software metrics - Software coupling of experiment data collection and processing components
    a. Questions:
        i. Are there any defined data formats or schemas that any of the components rely on for communication? (i.e. XML, CSV, XLS, JSON, Proprietary, etc)
        ii. Alternatively, is each connection between components that share any data uniquely configured?
        iii. Are there any instances where more than one component references the same data resource (i.e. the same file is accessed by multiple components)?
        iv. If machine agents exist in the study, do they modify the representation of the task environment directly or pass a message to trigger an action?
    b. Description: The degree to which distinct components of the data collection architecture are interdependent on each other.
    c. Research Value: Use the responses to assess where along the coupling spectrum the specific study's data architecture lies. A common theme of a high degree of coupling between the distinct research studies suggests that an effort to provide a higher level of abstraction (encapsulation of the details that change) will have a positive effect.
    d. Examples:
        i. Machine agents that take action in the task must modify the state of the task, and can do so on the spectrum of directly manipulating the internal representation of the task to sending a generic message of an action in the same standard input format as a human subject.

ii. A cognitive model (an algorithm, discrete event simulator, cognitive architecture, etc) may require physiological observations in addition to updates about the task environment state to make assessments, predictions and/or recommend actions.

2. Standard software metrics - Software cohesion of experiment components (data collection and processing)
    a. Questions:
        i. Which components of the study collect and process any data streams?
        ii. Which components contain any logic that forwards the data to another component in the study?
    b. Description: A measure of how organized or centralized is the logic that is used to manage the collection and processing of the data collected during an experiment's trial. The spectrum of cohesion with respect to these studies ranges from high cohesion logic used to route the experiment's data may be centralized and grouped according to functionality down to low cohesion where functionality is grouped arbitrarily.
    c. Research Value: The responses will be used to assess where on the cohesion spectrum the specific study's data architecture lies. A low degree of cohesion is likely in an architecture that was not intentionally designed to efficiently manage data and built in an ad hoc fashion. A common theme of low cohesion suggests that and effort to standardize and consolidate functionality will improve researchers' ability to maintain and extend the software architectures of their experiments. The primary benefit could be gained by reducing necessary time and personnel resources for building, maintaining and operating the experimental data processing architecture.
    d. Examples:
        i. All data output (such as: human subject actions and performance, model workload level assessment, task environment updates) is sent to a common component which contains the logic that distributes the data throughout the system appropriately.
        ii. Alternatively, each component is configured independently with respect to where its output is sent. Such as, the physiological observations are sent to a model predicting workload, the task environment that displays the data and a storage component for post-trial analysis. In this case, if a different physiological hardware with a new format is used, all three components it is connected to (task, model and storage) will likely also need to be updated.
3. Cyclomatic complexity – Data flow process
    a. Questions:

i. List each component of the study that either produces or consumes data. Additionally, include the number of distinct data streams (data originating from a distinct source) that enter and leaved each component.
ii. Describe each step of processing that occurs on a stream of data starting from its original production. Repeat for each data stream source that is produced or consumed during the experiment.

b. Description: Cyclomatic complexity refers to the count of the linearly independent path through program code or a system. The complexity measure is measured using a tool called a control flow graph (a directed graph). Each node of the graph is an indivisible group of commands.
c. Research Value: Gain insight into the complexity due to the paths the data must travel independent of the makeup of the details of the component that the data passes through.
d. Examples:
    i. Physiological observations of the human participant's heart rate is sent to a model estimating workload, a data visualization element that displays to the task environment, and thirdly to persistent storage. Each of the connections of physiological data to the other components would be the start of a distinct path.

4. Cyclomatic complexity – Synchronization complexity
    a. Questions:
        i. Do any of the data collection components of the study (data storage or processing) require temporal alignment of the received data?
    b. Description: Synchronization complexity is cyclomatic complexity except that it applies to the interleaving of multiple concurrent threads rather than a single thread of operations. Thus, this metric would only apply when an experiment is actively using the data streams from multiple, disparate sources combined into a single data consumer.
    c. Research Value: This metric can identify additional sources of complexity not indicated by the standard, single-thread control flow graph measure.
    d. Examples:
        i. Task performance (instantaneous) and physiological measures being used by a computational model to predict human subject workload.
        ii. Logging multiple data streams produced from disparate source (i.e. physiological measures, model output and participant actions) into a single, common storage or processing source.
5. Algorithmic complexity – Big-O message communication message complexity
    a. Questions:
        i. For all the components that produce any data, do they specify within their internal configuration where their outputted data is

sent within the data architecture? Alternatively, do all components broadcast their output to all other components of the study?

ii. Are there any components within the data architecture that contains logic specifically used to control the flow of data from other components?

1. If so, how is this component connected to the other data components?

b. Description: This metric refers to the growth of the total number of message packets that are required as the number of components in the experiment (n) is increased. An architecture in which every component is connected to every other component (such as in a basic data bus), will have a message growth rate of n^2. Even if not every component in the system is both a producer and consumer, $f((n-1)^2) = O(n^2)$.

c. Research Value: This measure can reveal complexity in an architecture that is not designed to efficiently handle the addition of many components that both produce and consume data streams, such as workload models based on physiological data. The existence of a high growth rate in an architecture may not create a severe enough impact to prevent its use at the moment; however, as research into building and testing multiple computational models and agents increases, the impact will become an unavoidable issue.

d. Examples:

i. A study contains several physiological measures being collected off the human participant to include heart rate, heart rate variability and several EEG locations and frequency bands. Currently, all the physiological measures are sent to both a computational model producing workload estimates off of task performance and all physiological measures in addition being sent to the task environment for display. The original architecture is built so that each component sends its output to all the other components which either consume or ignore the data. If an additional computational model is added, say to use another method of estimating workload from physiological measures and task performance, all other $n-1$ streams of data are duplicated in order to send their data to and from this new component.

6. General complexity measures – Data flow process description

a. Questions:

i. What data collection capabilities are essential in order for you to perform your study? Examples:

1. Collecting human subject actions/events in the task environment
2. Storing full resolution observations from physiological sensors

3. Ability for a machine agent to interact with the task environment simultaneously with any other human or machine agents

ii. What data collection limitations are preventing you from desired research activities? Examples:

1. Analysis of consolidated subject actions & physiological sensors in real-time

b. Description: This measure is a natural language description of the processes that must take place to capture, store and prepare data for analysis with regard to a specific experiment's data architecture.

c. Research Value: This description can be used to identify common bottlenecks regarding the manual administration of experimental data that may be alleviated by a product such as a software framework.

7. General complexity measures – Data architecture maintenance
   a. Questions:
      i. How many data collection components and data collection architectures used in this study are from a previous study?
         1. Of those, how many are from your own group/organization?
         2. Would you have used an existing component if there was less of a time or other resource barrier to implementation?
      ii. What, if any, defined process did you follow for building the architecture to collect the data produced during this study?
      iii. What, if any, current data collection capability do you see as a limitation to the research activities you would like to perform?
      iv. How many distinct programming languages were used in this study? Such as, JavaScript, C#, Java, Python, etc.
   b. Description: These are activities required to build and maintain the data collection software architecture.
   c. Research Value: Intricate/involved processes necessary to modify and update components of a data collection software architecture indicate that additional, unnecessary resources are being spent on activities that do not directly improve the results of the experiment itself.
8. General complexity measures – Post-trial process requirements
   a. Questions:
      i. What post-trial analysis of the data was conducted?
      ii. What actions had to be taken in order to arrange the collected data so that it could be analyzed?
      iii. Are there instances where an automated process or standard format would have reduced the time/effort of arranging the data for analysis?

Appendix B: Data Infrastructure Query Results

This portion of the index includes the results from the SME and IRB protocol experimental design reviews.

SME Interview 1

Background	
1	The study was conducted in 2007 as part of a graduate degree program. The overall goal was to observe EEG in correspondence with levels of expertise. The task required the human participant to push a single button in response to a stimulus that appeared on a computer screen. EEG observations were recorded from the human participant while the stimulus was produced, during which time the participant would hit a button.
2	The task environment was one source of data which data was collected on which was the timing and sequence of the stimuli presented to the participants. A second set of collected data were EEG signals from a passive scalp device. A third set of data collected was button press information (which one and timing). A questionnaire capturing basic demographic information was also collected. Finally, a video camera captured the trial for reference
3	All of the data was processed and stored on one machine. One path of the process, a MATLAB program handled driving the events to display the stimulus to the participant and capture their respective button pushes. A second path, the EEG signals were passed through an amplifier and recorded on the machine.
Questions	
1-i	The stimuli sequences and button pushes were stored in a MATLAB file format in a matrix. The EEG signals were also stored in the same manner.
1-ii	The two MATLAB programs communicated with each other using functions. The purpose of the communication was to trigger the recording of EEG signals right before a stimulus occurred and until a button was pushed so it would not be constantly running and cause the machine to run out of memory.
1-iii	No
1-iv	The study did not have any computational agents. If it did, an agent would modify the timing of the stimuli to keep the participant in a certain range. This would require the agent to send some sort of offset or sequence adjustment for producing the stimuli.
2-i	The two “components” that collect data are the two MATLAB programs. One that collects the sequence and button press data, and another that collects the EEG signal data.
2-ii	The main trial script will trigger the EEG collection. The main MATLAB script does not forward the data, but it does process the participant timings to assess performance for analysis post-trial.
3-i	EEG -> MATLAB program #2 Participant actions -> Task driving MATLAB code -> stored in MATLAB file for post process Task event sequences -> MATLAB file for post processing

3-ii	No active processing occurred on the data. One way this could have occurred in this study would have been: Button presses and timings -> Verification of accuracy -> Display metric back to participant
4-i	The only alignments that needed to occur in this study were the two matrices. One that held the stimulus occurrence timings and participant button presses and the matrix that held the EEG observations.
5-i	The modules in this study do not send their data externally with the exception of the task MATLAB code which triggers the recording of EEG.
5-ii	No, data is shared, but not control.
6-i	Time stamps of button presses and button types. Sequences of the stimuli need to be captured in sequence with the button presses. There must also be the same sequencing for the stimuli and button press events correlated to the recordings of EEG.
6-ii	The amount of EEG signals that can be captured at one time was limited by the computer's memory.
7-i	The MATLAB code for producing the stimuli and capturing button presses was modified code. Additionally, the EEG calibration and tuning functions were reused from previous experiments.
7-ii	The reused pieces came from the lab that the experiment was run from.
7-iii	The MATLAB Toolbox is a set of software tools that were used to build the code for the experiment data capture, but no design standards were explicitly used.
7-iv	The head-mounted eye tracking hardware interfered with the EEG signal collection and could not be used. There was a hardware limit of memory available for storing EEG signals.
8-i	One was used, MATLAB.
8-ii	The behavioral analysis consisted of investigating the response times of the participant. Event-related potentials (ERP) and a linear discriminant classifier were used for analysis on the EEG waveforms.
8-iii	For EEG, the primary work was to remove noise (such as eye-blinks and other artifacts). Then the sequence of EEG captures needed to be time-aligned with the stimuli conditions and participant responses. The fact that the exact timings and sequence of the stimuli events was known a priori and was static made the alignment easy. One matrix could be directly combined with another.
8-iii	The automation of artifact removal from the EEG signals would have been a huge time saver.

SME Interview 2

Background	
1	The study is being put together as of Jan 2016. It includes a handgun firing task focused on improving performance defined by Euclidean distance from the target. The human subject is connected with multiple physiological sensors that collect as the task is performed. The data is collected for off-line analysis in order to perform feature selection for the feature that has the greatest predictive power for performance (target accuracy). The vision for future studies is to use the models and

	features developed from this currently described study and have them fed with the physiological features in real-time.
2	The target in the task environment (firing range) was a source of data as the location of the hits from the bullets. There were multiple streams of data captured from physiological sensors. An audio queue was provided as the stimulus for the subject to fire the weapon. There was also a video log of the subject and view down the range to the target. The following physiological sources were collected from sensors. A Zephyr BioHarness was used to collect Heart Rate, ECG and posture. Toby Glasses were used to track eye position and movement. Portable EEG was used to capture 13 channels of EEG.
3	Due to hardware limitations, the data from each the Zephyr BioHarness, Toby Glasses and Portable EEG, were stored on different devices. Otherwise, all of the data would be processed and stored on a single machine. All of the data is processed post-trial, with the exception that the Portable EEG output can be viewed in real-time.
Questions	
1-i	The data is not transferred in real-time. And if it is in a subsequent study, there is no current decision for all the data to be of a particular type. However, it would most likely be stored in JSON format as other tools within the lab already use that convention.
1-ii	N/A, no connection between experimental modules.
1-iii	No
1-iv	The study did not have any computational agents.
2-i	There is one machine that collects the EEG signal data.
2-ii	No
3-i	Each of the physiological sensors produce a distinct data stream.
3-ii	None of the streams are processed, rather, they flow directly to storage with the exception of EEG with is sent to a monitor to view the current status of the streams.
4-i	No
5-i	All of the components store data right to persistent storage.
5-ii	No
6-i	Physiological sensor data should remain at as high of a sampling resolution as possible.
6-ii	None at this time.
7-i	Zephyr and EEG processing software.
7-i-1	The reused pieces came from the lab that the experiment was run from.
7-i-2	No, the available resources were easy enough to modify in order to work for this study.
7-ii	None
7-iii	None for this study, maybe an issue with image analysis of the video pointing at the

	target.
7-iv	Proprietary software to gather physiological sensor data and R and JavaScript to align data post trial.
8-i	Feature selection for predicting accuracy.
8-ii	Put through RAVED system.
8-iii	Would save time spent aligning data.

SME Interview 3

<table>
<tr><td>Background</td><td></td></tr>
<tr><td>1</td><td>The study will be conducted in order to compare multiple methods of workload estimation on the same task captured from the same subject at one time. The data streams are composed of both behavioral and physiological data types.</td></tr>
<tr><td>2</td><td>The study collects behavioral, physiological and demographic data. The behavioral data is the performance data on the task (MATB). The following physiological sources were collected from sensors. A Zephyr BioHarness was used to collect Heart Rate, ECG and posture. Toby Glasses were used to track eye position and movement. Portable EEG was used to capture 13 channels of EEG.
<table>
<tr><td>Performance Data</td><td>Stored in database by the MATB task software.</td></tr>
<tr><td>Zephyr BioHarness</td><td>Data is stored in the “puck”, which a piece of the equipment.</td></tr>
<tr><td>Portable EEG</td><td>Data is saved to a folder on the device (computer/tablet) that the amplifier and signal digitizer is connected to.</td></tr>
<tr><td>Toby Glasses</td><td>Recorded to SD card in the hardware</td></tr>
<tr><td>Demographic data</td><td>Saved to folder on a separate machine the surveys are taken on.</td></tr>
<tr><td>Subjective Workload (NASA TLX)</td><td>On same machine as MATB task, but separate program.</td></tr>
</table></td></tr>
<tr><td>3</td><td>Due to hardware limitations, the data from each the Zephyr BioHarness, Toby Glasses and Portable EEG, were stored on different devices. Otherwise, all of the data would be processed and stored on a single machine. No modules in the experiment took data from another module during the course of the experiment.

All of the data is processed post-trial, with the exception that the Portable EEG output can be viewed in real-time. Once all sources are converted into CSV, then they are all ingested into a database using the local lab developed program to align the data from the disparate sources all to a common time series.
<table>
<tr><td>Performance Data</td><td>Pulled out of database and exported to CSV.</td></tr>
<tr><td>Zephyr BioHarness</td><td>Collected data is processed through AFRL software into a CSV (captured at</td></tr>
</table></td></tr>
</table>

		250 Hz for ECG, 25 Hz respiration rate, 100 Hz position accelerometer, 1Hz posture)
	Portable EEG	Sample rate is 128 Hz, data processed through AFRL software
	Toby Glasses	Uses proprietary process to convert collected data to a CSV.
	Demographic data	Traits are stored as the averages of scores across all surveys.
	Subjective Workload (NASA TLX)	On same machine as MATB task, but separate program.
Questions		
1-i	No, the data is stored in proprietary formats.	
1-ii	No	
1-iii	No real-time models exist in the study.	
2-i	The task environment creates a performance data stream stored directly to a database. The physiological sensors create data streams that are stored directly to disparate data storage locations.	
2-ii	None, the data stays locally for each module.	
3-i	Task environment performance measures are streamed to a database (persistent storage). EEG sensor data is visualized on the machine (desktop/tablet) for visualization for verification that it is recording correctly.	
3-ii	The qualitative data, condition descriptions and demographics data are coded for digital storage in the database. Otherwise, all other data is stored as already described.	
4-i	Only the task environments has small elements that must be aligned within itself. This includes task events, such as the occurrence of stimuli, being aligned to the participant's actions, key presses and mouse clicks.	
5-i	The data is stored locally to each module of the experiment. Even the task has its data stream stored within its own software.	
5-ii	No	
6-i	Collect as high resolution data as the hardware allows.	
6-ii	Nothing, limiting the experimental design, but it would make it easier to verify and align if all of the data sources (physio sensor hardware) could stream externally to the device.	
6-iii	No roadblocks	
7-i	Zephyr data processing software. Demographic LIME survey tool. The task environment, JavaScript version of modifiable MATB (mMATB). Each of these came from previous work.	
7-ii	Consistent naming conventions for files and folders. This helps to keep the subjects and trial (conditions) straight.	
7-iii	CSV file size limitations. Some of the data from physiological sensors for a single trial exceeds the size limitation of CSV files.	

8-i	Comparisons between measures.
8-ii	The data must be run through RAVED. Has to be converted to a readable format by RAVED (CSV files). Demographic data is run through an R script.
8-iii	Generating common timestamps. Having a synchronization port on the hardware devices to sync to a common clock at the beginning of a trial.

SME Interview 4

<table>
<tr><td>Background</td><td></td></tr>
<tr><td>1</td><td>This study is currently ongoing as of the interview on 13 Jan 2016. Subjects are still being run through the task to collect more data. The study is focused on collecting data regarding the “Augment” piece of the Sense-Assess-Augment cycle. The researchers are doing this by including task augmentation during the trials that turn on according to a timed schedule during the task. The augmentation is not controlled by computational models yet because the models are not accurate enough to effectively manage the automation. The actual task description is in</td></tr>
<tr><td>2</td><td><table>
<tr><td>Performance Data</td><td>Created by the task environment</td></tr>
<tr><td>BioRadio (EEG)</td><td>EEG</td></tr>
<tr><td>SmartEye</td><td>Off person pupilometery
Vertical EOG
Horizontal EOG</td></tr>
<tr><td>BioHarness</td><td>Respiration Rate</td></tr>
<tr><td>Microphone</td><td>Voice stress analysis</td></tr>
<tr><td>Subjective Measures</td><td>NASA TLX (between rounds of trial according to the schedule in the appendix)</td></tr>
<tr><td>Computational Models</td><td>There are 18 models that accept various data features and produce outputs, predicting workload or performance</td></tr>
</table></td></tr>
<tr><td>3</td><td>All of the data flows in real-time to a Universal Data Bus that resides on a computer in the lab network. The software for the data bus was developed collaboratively by the researchers involved in the study and an external research and software development company also involved in executing the study.</td></tr>
<tr><td>Questions</td><td></td></tr>
<tr><td>1-i</td><td>Data is sent from the sensors to the data bus as raw data and then is sent in XML format to the computational models or other modules.</td></tr>
<tr><td>1-ii</td><td>No, connections are managed by the data bus.</td></tr>
<tr><td>1-iii</td><td>Unknown, but unlikely.</td></tr>
<tr><td>1-iv</td><td>The models take in performance data from the task environment, but none of the models control the task environment. There is a script in the task schedule that toggles augmentation.</td></tr>
<tr><td>2-i</td><td>The computational models take physiological and performance features as input streams.</td></tr>
</table>

2-ii	The data bus accept raw sensor data and forward the digitized, XML format version of it out to the rest of the data workflow architecture.
3-i	There are 18 distinct computational models in the study. The all take a subset of the performance and physiological features.
3-ii	Raw physiological sensor data is sent to the Universal Data Bus. From there, it is exported as XML to the computational models.
4-i	Yes, they may receive multiple data feature streams.
5-i	Yes, the data bus is configured to send the data to specific models over the network LAN.
5-ii	Yes, the data bus.
5-ii-1	Over web sockets on the LAN.
6-i	Physiological sensor data should remain at as high of a sampling resolution as possible. Network Time Protocol is available on the network.
6-ii	None at this time.
7-i	The data bus is from the previous HUMAN Lab Formal Study 1 at least.
7-i-1	The data bus software was developed by one of the Primary Investigators.
7-i-2	Did not ask
7-ii	Did not ask
7-iii	None for this study, maybe an issue with image analysis of the video pointing at the target.
7-iv	Did not ask. Postgress SQL database to store collected data.
8-i	Feature selection for predicting to analyze computational model performance.
8-ii	The data is aligned with the central data bus in real-time and then stored into a SQL database that maintains that alignment.
8-iii	This is already achieved with the existing data bus software configuration.

IRB Protocol Review 1

Background	
1	This project will seek to define a robust method for remotely and noninvasively determining heart rate through application of imaging technology. It also seeks to better understand the relationship between heart rate measures and mental workload levels experienced by operators. Data will also be collected to determine which user tasks are discarded as mental workload levels increase. Subjects will interact with the Air Force Multi-Attribute Test Battery (AF_MATB), running on a laptop computer. The AF_MATB provides a method to manipulate an operator's task load and impose different levels (high, med, low) of mental workload. The original MATB software has become a mainstay for psychological research regarding cognitive workload and this version has simply updated the software to be compatible with modern operating systems.Subjects will use the standard laptop keyboard in addition to a USB joystick to perform the given tasks. The task does not

	depend on real-time and there is no automation.
2	<table><tr><th>Equipment</th><th>Measurement</th></tr><tr><td>Stored by the AF MATB task software</td><td>Task performance Data</td></tr><tr><td>NASA TLX</td><td>Subjective workload assessment</td></tr><tr><td>Analytical Spectral Devices (ASD) FieldSpec® Pro spectrometer</td><td>Measures spectral bands that are associated with heart rate</td></tr><tr><td>BIOPAC 150 with ECG amplifier</td><td>electrical signals associated with the beat of the human heart</td></tr></table>
3	<table><tr><td>Performance Data</td><td>Recorded by MATB software</td></tr><tr><td>Spectral wavelengths</td><td>Processed by proprietary software to MATLAB readable format</td></tr></table>
Questions	
1-i	The data is not transferred in real-time.
1-ii	N/A, no connection between experimental modules.
1-iii	N/A, no connection between experimental modules.
1-iv	N/A, the study did not have any computational agents.
2-i	No modules collect any external data streams.
2-ii	None
3-i	Data streams: ECG, performance data, spectral wavelengths. None of the streams are communicated external to the module that created them.
3-ii	N/A the streams are not processed in real-time.
4-i	No
5-i	All of the components store data directly to distinct persistent storage mechanisms.
5-ii	No
6-i	Physiological sensor data should remain at as high of a sampling resolution as possible.
6-ii	Not discussed
7-i	The task environment is the same and collects performance data.
7-i-1	The reused task environment is from the same organization.
7-ii	Unknown
7-iii	Unknown
7-iv	Unknown
8-i	The data from the ECG will also be analyzed to determine whether significant changes in heart rate or heart rate variability occurred during each experimental session and these values will be correlated with changes in the reflectance data collected from the ASD.
8-ii	Group data collections by trial.

8-iii	Unknown

IRB Protocol Review 2

<table>
<tr><td>Background</td><td></td></tr>
<tr><td>1</td><td>The aim of this study is to determine the effect of localized temperature changes on vigilance performance and whether individual stress appraisals moderate the relationship between localized temperature changes and vigilance performance. Electrocardiography, electrooculography, and cerebral oximetry data will be measured during a vigilance task to determine the relationship between these physiological measures, temperatures changes, and human performance. The task does not depend on real-time and there is no automation.</td></tr>
<tr><td>2</td><td><table>
<tr><th>Equipment</th><th>Measurement</th></tr>
<tr><td>Air Traffic Control Vigilance Task (Super Duper Lab)</td><td>Performance results (accuracy)</td></tr>
<tr><td>Cerebral Oximeter (CO)</td><td>Noninvasive blood oxygen saturation</td></tr>
<tr><td>BIOPAC 150 with EOG amplifier</td><td>EOG, eye positions</td></tr>
<tr><td>BIOPAC 150 with ECG amplifier</td><td>ECG, electrical signals associated with the beat of the human heart</td></tr>
</table></td></tr>
<tr><td>3</td><td><table>
<tr><td>Performance Data</td><td>Recorded by Super Duper Lab software</td></tr>
<tr><td>Physiological measures</td><td>The physiology data is filtered to remove any extraneous data or outliers</td></tr>
</table></td></tr>
<tr><td>Questions</td><td></td></tr>
<tr><td>1-i</td><td>Not specified.</td></tr>
<tr><td>1-ii</td><td>It appears that all the data collecting modules are connected in some fashion. It is unknown what the ‘single process’ phrase refers to, task procedure or computational instance of software. This is a quote from the protocol: “Additionally, BIOPAC software will continue to record all of the ECG and EOG data and INVOS software for the CO data. The temperature changes, performance data, and physiological data will all be coupled within a single process to accurately keep the time scale consistent throughout the experiment.”</td></tr>
<tr><td>1-iii</td><td>No, the data is passing in one direction, only being saved.</td></tr>
<tr><td>1-iv</td><td>N/A, the study did not have any computational agents.</td></tr>
<tr><td>2-i</td><td>No modules collect any external data streams.</td></tr>
<tr><td>2-ii</td><td>None</td></tr>
<tr><td>3-i</td><td>Data streams: CO, EOG, ECG all flow one-way into storage.</td></tr>
<tr><td>3-ii</td><td>N/A the streams are not processed in real-time.</td></tr>
<tr><td>4-i</td><td>No, alignment occurs after the completion of the experiment.</td></tr>
<tr><td>5-i</td><td>Unknown, unclear from protocol equipment description.</td></tr>
<tr><td>5-ii</td><td>No</td></tr>
</table>

6-i	Physiological sensor data should remain at as high of a sampling resolution as possible.
6-ii	Not discussed
7-i	Unknown
7-i-1	Unknown
7-i-2	Unknown
7-ii	Unknown
7-iii	Unknown
7-iv	Unknown
8-i	To test whether localized temperature changes have an effect on vigilance performance one-way Analysis of Variance, with temperature as the independent variable and performance as the dependent variable will be conducted. To test whether individual stress appraisals moderate the relationship between localized temperature changes and vigilance performance a hierarchical regression will be conducted.
8-ii	The sources had to be connected at the beginning of the trial.
8-iii	Unknown

IRB Protocol Review 3

Background	
1	If the results from the previous experiment show that vigilance performance does improve when localized temperature conditions change, then an additional experiment will be conducted. Experiment 2 will investigate whether performance can be additionally improved by using real-time physiological data to predict when vigilance is declining and implement a localized temperature change at that time.The aim of this study is to determine the effect of localized temperature changes on vigilance performance and whether individual stress appraisals moderate the relationship between localized temperature changes and vigilance performance. Electrocardiography, electrooculography, and cerebral oximetry data will be measured during a vigilance task to determine the relationship between these physiological measures, temperatures changes, and human performance. The task does not depend on real-time and there is no automation.
2	Equipment / Measurement Air Traffic Control Vigilance Task (Super Duper Lab) — Performance results (accuracy) Cerebral Oximeter (CO) — Noninvasive blood oxygen saturation BIOPAC 150 with EOG amplifier — EOG, eye positions BIOPAC 150 with ECG amplifier — ECG, electrical signals associated with the beat of the human heart
3	Performance Data — Recorded by Super Duper Lab

	<table><tr><td></td><td>software</td></tr><tr><td>Physiological measures</td><td>The physiology data is filtered to remove any extraneous data or outliers</td></tr></table>
Questions	
1-i	Not specified
1-i-1	If the results from the previous experiment show that vigilance performance does improve when localized temperature conditions change, then an additional experiment will be conducted. Experiment 2 will investigate whether performance can be additionally improved by using real-time physiological data to predict when vigilance is declining and implement a localized temperature change at that time.The aim of this study is to determine the effect of localized temperature changes on vigilance performance and whether individual stress appraisals moderate the relationship between localized temperature changes and vigilance performance. Electrocardiography, electrooculography, and cerebral oximetry data will be measured during a vigilance task to determine the relationship between these physiological measures, temperatures changes, and human performance. The task does not depend on real-time and there is no automation.
1-i-2	Unknown, the configuration of the vigilance assessment model, physiological data and thermoelectric pad and blanket.
1-ii	N/A, the study did not have any computational agents.
2-i	ECG, EOG and CO data are streamed to a source that checks for a predetermined vigilance decrement. However, the method for evaluating the trigger condition was not established yet. The temperature control will have to be triggered
2-ii	Unknown
3-i	Data streams producers: CO, EOG, and ECG. Data streams consumer: temperature controller (either decision is made externally and receives an instruction to change temperature or decision is made internally and receives each of the physiological streams).
3-ii	Unknown, the connection design was not presented.
4-i	Real-time temporal alignment will be necessary if the logic making the vigilance level assessment uses a combination of physiological measures. It may be the case that each of the streams has their own trigger threshold and they do not need to be aligned with each other.
5-i	Unknown, unclear from protocol equipment description.
5-ii	Unknown, unclear from protocol equipment description.
6-i	Physiological data streams need to be collected and assessed as quickly as possible in order to match the assessment to the real-world.
6-ii	Not discussed
7-i	The process for collecting all of the data streams into one location.
7-i-1	Unknown
7-i-2	Unknown
7-ii	Unknown

7-iii	Unknown
7-iv	Unknown
8-i	The physiology data will be analyzed as in Experiment 1, but with an additional real time component to ideally predict the vigilance decrement and counter it in real time. Performance data will be recorded and analyzed then compared to results from Experiment 1 to determine how the effects of temperature changes initiated from physiology signals differ from one at set time interval.
8-ii	The sources had to be connected at the beginning of the trial.
8-iii	Unknown

Appendix C: Meta-Study

There are two sections of this appendix, publication list and coded results. The first section provides the citation for each of the published experiments used in the meta-study. The numbered list indicates the identifier for the study which corresponds to the same ID field in the coded results. Following the list of studies are the coded results.

Measure	Measurement Range
Year Published	Integer: 1996 – 2015
Count Modules Producing Data Streams (*P*)	Integer: $P \geq 0$
Count Modules Consuming Data Streams (*C*)	Integer: $C \geq 0$
Count Total Number of Distinct Modules (*T*)	Integer: $T \geq 0,\ T \leq P + C$
Real-Time Inter-Module Communication	{Yes, No}
Data Standards Mention	{Yes, No}

Dynamism of Automation	{Static, Dynamic, None}
Adaptive Automation Exists	{Yes, No}

The total number of distinct modules is obtained by the number of individual modules according to the taxonomy. Since some modules may both produce and consume data streams, they are counted in both *P* and *C*, but only once in *T*. Thus, the total number of distinct modules may be less than the sum of consuming and producing modules. Real-Time Inter-Module communication is considered "Yes" if there exists at least one stream that is sent from a producing module to a separate, distinct consuming module which processes the stream and may or may not export the result. The most common example of this is a computational model predicting workload from one or more physiological data streams.

Data standards mentioned is coded as "Yes", when the authors mention details of the data infrastructure configuration. The threshold for coding a "Yes" is at least mentioning what specific type of hardware is used for physiological or behavioral measures.

Dynamism of Automation has three levels: None, Static and Dynamic. "None" is coded when there is no automation that assists the human perform the task. "Static" is coded when there is computational assistance of the same task the human is performing, but the automation is configured to be on or off over the course of each entire trial (between experimental conditions). "Dynamic" is coded when there is computational assistance, and the type of automation changes during the course of a trial (within an experimental condition). The types of changes include varying the level of automation or triggering the automation on or off in real-time.

Each of the publications was reviewed for data architecture limitations that were either explicitly expressed by the authors or implied from the design. For example, it could be implied that an experiment had no ability to perform online data analysis across all physiological measures if each of the data streams were saved to unconnected storage devices.

Publications Included in the Meta-Study

Note: Two of the included studies describe multiple distinct experiments that occurred at different points in time. Each study is listed only once and multiple ID's of experiments are all listed next to a single publication where applicable. The experiments that came from the same publication are 23, 27 & 28 and 32 & 33.

1. Brookings, J B, Glenn F. Wilson, and C R Swain. "Psychophysiological Responses to Changes in Workload during Simulated Air Traffic Control." *Biological Psychology* 42.95 (1996): 361–377. Web.
2. Parasuraman, Raja, M Mouloua, and R Molloy. "Effects of Adaptive Task Allocation on Monitoring of Automated Systems." *Human factors* 1996: 665–679. Web.
3. Hoc, Jean-Michel, and Marie-Pierre Lemoine. "Cognitive Evaluation of Human-Human and Human-Machine Cooperation Modes in Air Traffic Control." *International Journal of Aviation Psychology* 8.1 (1998): 1–32. Web.
4. Morgan, C, CC Cook, and C Corbridge. *Dynamic Function Allocation for Naval Command and Control*. N.p., 1999. Web. 7 Jan. 2016.
5. Endsley, M R, and D B Kaber. "Level of Automation Effects on Performance, Situation Awareness and Workload in a Dynamic Control Task." *Ergonomics* 42.3 (1999): 462–492. Web. 7 Jan. 2016.
6. Itoh, M., G. Abe, and K. Tanaka. "Trust in and Use of Automation: Their Dependence on Occurrence Patterns of Malfunctions." *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028).* Vol. 3. IEEE, 1999. 715–720. Web. 7 Jan. 2016.
7. Kaber, DB, E Onal, and MR Endsley. "Design of Automation for Telerobots and the Effect on Performance, Operator Situation Awareness, and Subjective

Workload." *Human Factors and Ergonomics in Manufacturing* 10.4 (2000): 409–430. Web. 7 Jan. 2016.
8. Russell, Chris A., and Glenn F. Wilson. *Application of Artificial Neural Networks for Air Traffic Controller Functional State Classification*. N.p., 2001. Web. 11 Jan. 2016.
9. Smith, Michael E. et al. "Monitoring Task Loading with Multivariate EEG Measures during Complex Forms of Human-Computer Interaction." *Human Factors* 43.3 (2001): 366–380. Web.
10. Lorenz, Bernd et al. "Automated Fault-Management in a Simulated Spaceflight Micro-World." *Aviation, space, and environmental medicine* 73.9 (2002): 886–97. Web. 6 Jan. 2016.
11. Clamann, M. P., and D. B. Kaber. "Authority in Adaptive Automation Applied to Various Stages of Human-Machine System Information Processing." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 47.3 (2003): 543–547. Web. 7 Jan. 2016.
12. Wilson, Glenn F., and Christopher A. Russell. "Real-Time Assessment of Mental Workload Using Psychophysiological Measures and Artificial Neural Networks." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 45.4 (2003): 635–643. Web. 19 Jan. 2015.
13. Freeman, Frederick G. et al. "An Evaluation of an Adaptive Automation System Using a Cognitive Vigilance Task." *Biological Psychology* 67.3 (2004): 283–297. Web. 3 Dec. 2015.
14. Dorneich, Michael et al. "Closing the Loop of an Adaptive System with Cognitive State." *Proceedings of the Human Factors and Ergonomics Society Conference*. New Orleans, LA: N.p., 2004. 5. Web. 12 Jan. 2016.
15. Dorneich, MC, and S Mathan. "Enabling Improved Performance Though a Closed-Loop Adaptive System Driven by Real-Time Assessment of Cognitive State." *Foundations of Augmented Cognition* (2005): n. pag. Web. 12 Jan. 2016.
16. Fairclough, Stephen H, and Louise Venables. "Prediction of Subjective States from Psychophysiology: A Multivariate Approach." *Biological psychology* 71.1 (2006): 100–10. Web. 31 Dec. 2015.
17. Poythress, Marianne et al. "Correlation between Expected Workload and EEG Indices of Cognitive Workload and Task Engagement." *Foundations of Augmented Cognition* 1 (2006): 32–44. Web.
18. Dorneich, M. C., P. M. Ververs, S. D. Whitlow, et al. "Neuro-Physiologically-Driven Adaptive Automation to Improve Decision Making Under Stress." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50 (2006): 410–414. Web.
19. Kaber, David B., Melanie C. Wright, and Mohamed A. Sheik-Nainar. "Investigation of Multi-Modal Interface Features for Adaptive Automation of a Human–robot System." *International Journal of Human-Computer Studies* 64.6 (2006): 527–540. Web. 3 Dec. 2015.

20. Kaber, David B. et al. “Situation Awareness Implications of Adaptive Automation for Information Processing in an Air Traffic Control-Related Task.” *International Journal of Industrial Ergonomics* 36.5 (2006): 447–462. Web. 3 Dec. 2015.
21. Bailey, Nathan R. et al. “Comparison of a Brain-Based Adaptive System and a Manual Adaptable System for Invoking Automation.” *Human Factors: The Journal of the Human Factors and Ergonomics Society* 48.4 (2006): 693–709. Web. 7 Jan. 2016.
22. Linkens, Derek A. et al. “Towards on-Line Supervision and Control of Operational Functional State (OFS) for Subjects under Mental Stress.” *2007 IEEE/NIH Life Science Systems and Applications Workshop*. IEEE, 2007. 77–80. Web. 6 Jan. 2016.
23. Prevot, Thomas, Jeffrey Homola, and Joey Mercer. “Human-in-the-Loop Evaluation of Ground-Based Automated Separation Assurance for NextGen (AIAA).” *The 26th Congress of International Council of the Aeronautical Sciences*. N.p., 2008. Web. 7 Jan. 2016.
24. Mahfouf, M. et al. “Adaptive Fuzzy Approaches to Modelling Operator Functional States in a Human-Machine Process Control System.” *2007 IEEE International Fuzzy Systems Conference*. IEEE, 2007. 1–6. Web. 20 Nov. 2015.
25. Pattyn, Nathalie et al. “Psychophysiological Investigation of Vigilance Decrement: Boredom or Cognitive Fatigue?” *Physiology and Behavior* 93.1-2 (2008): 369–378. Web.
26. Ting, Ching-Hua et al. “Real-Time Adaptive Automation for Performance Enhancement of Operators in a Human-Machine System.” *2008 16th Mediterranean Conference on Control and Automation*. IEEE, 2008. 552–557. Web. 6 Jan. 2016.
27. - 28. Prevot, Thomas, Jeffrey Homola, and Joey Mercer. “Human-in-the-Loop Evaluation of Ground-Based Automated Separation Assurance for NextGen (AIAA).” *The 26th Congress of International Council of the Aeronautical Sciences*. N.p., 2008. Web. 7 Jan. 2016.
29. Cosenzo, Keryl et al. “The Effect of Appropriately and Inappropriately Applied Automation for the Control of Unmanned Systems on Operator Performance.” Army Research Laboratory Report, 2009. Web. 7 Jan.
30. Taylor, G. et al. “Comparison of Multiple Physiological Sensors to Classify Operator State in Adaptive Automation Systems.” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 54.3 (2010): 195–199. Web.
31. Ting, Ching-Hua Hua et al. “Real-Time Adaptive Automation System Based on Identification of Operator Functional State in Simulated Process Control Operations.” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 40.2 (2010): 251–262. Web. 20 Nov. 2015.
32. - 33. Cannon, Jordan A. et al. “An Algorithm for Online Detection of Temporal Changes in Operator Cognitive State Using Real-Time Psychophysiological Data.” *Biomedical Signal Processing and Control* 5.3 (2010): 229–236. Web. 26 June 2015.
34. Cosenzo, K. et al. “Adaptive Automation Effects on Operator Performance during a Reconnaissance Mission with an Unmanned Ground Vehicle.” *Proceedings of the*

Human Factors and Ergonomics Society Annual Meeting 54.25 (2010): 2135–2139. Web. 7 Jan. 2016.
35. Prinsloo, Gabriell E. et al. "The Effect of Short Duration Heart Rate Variability (HRV) Biofeedback on Cognitive Performance during Laboratory Induced Cognitive Stress." *Applied Cognitive Psychology* 25.5 (2011): 792–801. Web.
36. Dorneich, M. C., B. Passinger, et al. "The Crew Workload Manager: An Open-Loop Adaptive System Design for Next Generation Flight Decks." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 55.1 (2011): 16–20. Web.
37. Tango, Fabio et al. "Automation Effects on Driver's Behaviour When Integrating a PADAS and a Distraction Classifier." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6777 LNCS (2011): 503–512. Web.
38. Dijksterhuis, Chris, Karel A Brookhuis, and Dick De Waard. "Effects of Steering Demand on Lane Keeping Behaviour, Self-Reports, and Physiology. A Simulator Study." *Accident Analysis and Prevention* 43.3 (2011): 1074–81. Web. 20 Dec. 2015.
39. de Visser, E., and R. Parasuraman. "Adaptive Aiding of Human-Robot Teaming: Effects of Imperfect Automation on Performance, Trust, and Workload." *Journal of Cognitive Engineering and Decision Making* 5.2 (2011): 209–231. Web. 7 Jan. 2016.
40. Beaumont, Alison et al. "Reduced Cardiac Vagal Modulation Impacts on Cognitive Performance in Chronic Fatigue Syndrome." *PLoS ONE* 7.11 (2012): n. pag. Web.
41. Baldwin, Carryl L., and B. N. Penaranda. "Adaptive Training Using an Artificial Neural Network and EEG Metrics for within- and Cross-Task Workload Classification." *NeuroImage* 59.1 (2012): 48–56. Web.
42. Kidwell, B. et al. "Adaptable and Adaptive Automation for Supervisory Control of Multiple Autonomous Vehicles." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 56.1 (2012): 428–432. Web. 6 Jan. 2016.
43. Dorneich, M. C., P. M. Ververs, S. Mathan, et al. "Considering Etiquette in the Design of an Adaptive System." *Journal of Cognitive Engineering and Decision Making* 6.2 (2012): 243–265. Web.
44. Berg-yuen, Pia E K, Siddhartha S Mehta, and Robert A Murphey. "Control of Human-Machine Interaction for Wide Area Search Munitions in the Presence of Target Uncertainty." *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference*. Boston, Massachusetts: IEEE, 2012. 195–196. Web.
45. Halverson, Tim, Brad Reynolds, and Leslie Blaha. "SIMCog-JS : Simplified Interfacing for Modeling Cognition - JavaScript." 2013: 39–44.
46. Durkee, Kevin et al. "Real-Time Workload Assessment as a Foundation for Human Performance Augmentation." Ed. DylanD. Schmorrow and CaliM. Fidopiastis. *Springer-Verlag Berlin Heidelberg 2013* 8027 (2013): 279–288. Web. Lecture Notes in Computer Science.
47. Gartenberg, Daniel et al. "Adaptive Automation and Cue Invocation: The Effect of Cue Timing on Operator Error." *Proceedings of the SIGCHI Conference on Human*

Factors in Computing Systems - CHI '13. New York, New York, USA: ACM Press, 2013. 3121–3130. Web. 7 Jan. 2016.
48. Allen, Andrew P., Tim J C Jacob, and Andrew P. Smith. "Effects and after-Effects of Chewing Gum on Vigilance, Heart Rate, EEG and Mood." *Physiology and Behavior* 133 (2014): 244–251. Web.
49. Hogervorst, Maarten a., Anne-Marie Brouwer, and J Erp. "Combining and Comparing EEG , Peripheral Physiology and Eye-Related Measures for the Assessment of Mental Workload." *Frontiers in Neuroscience* 8.October (2014): 1–14. Web.
50. Matthews, G. et al. "The Psychometrics of Mental Workload: Multiple Measures Are Sensitive but Divergent." *Human Factors* 57.1 (2014): 125–143. Web. 4 Apr. 2015.
51. Durkee, Kevin T. et al. "System Decision Framework for Augmenting Human Performance Using Real-Time Workload Classifiers." *2015 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision*. IEEE, 2015. 8–13. Web. 10 Jan. 2016.

Meta-Study Coded Results

ID	Year Published	Count Data Producers	Count Data Consumers	Total Module Count	Real-Time Inter-Module Communication (Yes/No)	Adaptive Automation Exists (Yes/No)	Dynamism Of Adaptive Automation (Static/Dynamic)	Data Architecture Limitations From Author (Yes/No)	Data Standards Mention (Yes/No)
1	1996	4	0	4	No	No	None	No	Yes
2	1996	1	2	2	No	Yes	Dynamic	No	No
3	1998	4	3	4	Yes	No	None	No	No
4	1999	3	1	4	No	Yes	Static	No	No
5	1999	3	1	3	No	Yes	Static	No	No
6	1999	1	0	1	No	Yes	Static	No	No
7	2000	3	1	3	No	Yes	Static	No	Yes
8	2001	5	2	6	No	Yes	Static	No	No
9	2001	3	1	4	No	No	None	No	No
10	2002	4	2	5	Yes	Yes	Static	No	Yes
11	2003	2	1	3	Yes	Yes	Dynamic	No	No
12	2003	6	3	7	Yes	Yes	Dynamic	No	No
13	2004	3	1	3	No	Yes	Static	No	No
14	2004	3	3	6	No	Yes	Dynamic	No	No
15	2005	3	2	6	No	Yes	Dynamic	No	No

16	2006	7	1	7	No	No	None	No	No
17	2006	5	2	6	Yes	No	None	No	Yes
18	2006	5	2	6	Yes	Yes	Dynamic	No	No
19	2006	3	1	3	No	Yes	Static	No	No
20	2006	3	2	4	Yes	Yes	Dynamic	Yes	No
21	2006	4	1	4	No	No	None	No	No
22	2007	5	3	7	Yes	Yes	Static	No	Yes
23	2007	3	1	4	No	No	Static	No	No
24	2007	5	1	6	No	Yes	Static	No	No
25	2008	4	1	5	Yes	No	None	No	No
26	2008	5	3	7	Yes	Yes	Dynamic	No	Yes
27	2008	3	1	4	No	No	Static	No	No
28	2008	3	1	4	No	Yes	Dynamic	No	No
29	2009	4	1	5	No	Yes	Static	No	No
30	2010	4	1	5	No	No	None	No	No
31	2010	3	3	3	Yes	Yes	Dynamic	No	No
32	2010	3	0	3	No	No	None	No	No
33	2010	3	1	4	Yes	No	None	No	No
34	2010	4	1	5	No	Yes	Static	No	No
35	2011	4	2	6	Yes	No	None	No	No
36	2011	4	1	5	No	Yes	None	No	No
37	2011	3	3	5	Yes	Yes	Dynamic	No	No
38	2011	4	0	4	No	No	None	No	No
39	2011	2	1	3	Yes	Yes	Dynamic	No	No
40	2012	3	1	3	Yes	No	None	No	No
41	2012	3	2	4	No	No	None	Yes	Yes
42	2012	2	1	3	No	Yes	Dynamic	No	Yes
43	2012	5	4	6	Yes	Yes	Dynamic	No	No
44	2012	4	3	7	No	Yes	Dynamic	No	No
45	2013	3	3	3	Yes	No	None	Yes	Yes
46	2013	4	1	5	Yes	Yes	Dynamic	No	Yes
47	2013	2	2	4	Yes	Yes	Dynamic	No	No
48	2014	4	1	5	Yes	No	None	Yes	No
49	2014	5	2	8	No	No	None	No	No
50	2014	7	1	8	No	No	None	Yes	No
51	2015	8	1	8	No	No	None	No	Yes

Bibliography

[1] C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press, 1977.

[2] G. Taylor, L. Reinerman-Jones, K. Cosenzo, and D. Nicholson, "Comparison of Multiple Physiological Sensors to Classify Operator State in Adaptive Automation Systems," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 54, no. 3, pp. 195–199, 2010.

[3] K. Durkee, A. Geyer, S. Pappada, A. Ortiz, and S. Galster, "Real-time Workload Assessment as a Foundation for Human Performance Augmentation," *Springer-Verlag Berlin Heidelb. 2013*, vol. 8027, pp. 279–288, 2013.

[4] C. Lebiere, E. Biefeld, R. Archer, S. Archer, L. Allender, and T. D. Kelley, "IMPRINT/ACT-R: Integration of a task network modeling architecture with a cognitive architecture and its application to human error modeling," *Proc. 2002 Adv. Simul. Technol. Conf. San Diego, CA, Simul. Ser.*, vol. 34, pp. 13–19, 2002.

[5] R. Budiu, "ACT-R Experimental Method," 2013. [Online]. Available: http://act-r.psy.cmu.edu/wordpress/wp-content/uploads/2012/09/method.gif. [Accessed: 14-Dec-2015].

[6] J. Allanson and S. H. Fairclough, "A research agenda for physiological computing," *Interact. Comput.*, vol. 16, no. 5, pp. 857–878, Oct. 2004.

[7] J. Allanson, "Supporting the Development of Electrophysiologically Interactive Computer Systems," Lancaster University, 2000.

[8] S. Jo, R. Myung, and D. Yoon, “Quantitative prediction of mental workload with the ACT-R cognitive architecture,” *Int. J. Ind. Ergon.*, vol. 42, no. 4, pp. 359–370, 2012.

[9] V. F. Mancuso and M. D. McNeese, “Effects of Integrated and Differentiated Team Knowledge Structures on Distributed Team Cognition,” *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 56, no. 1, pp. 388–392, 2012.

[10] E. N. Wiebe, E. Roberts, and T. S. Behrend, “An examination of two mental workload measurement approaches to understanding multimedia learning,” *Comput. Human Behav.*, vol. 26, no. 3, pp. 474–481, May 2010.

[11] F. T. Eggemeier, G. F. Wilson, A. F. Kramer, and D. L. Damos, “Workload assessment in multi-task environments,” in *Multiple Task Performance*, London: CRC Press, 1991, pp. 207–215.

[12] G. I. Seffers, “Collaboration Research Puts the ’ I ’ in Team,” *Signal (AFCEA)*, p. 1, Aug-2015.

[13] J. M. Bindewald, M. E. Miller, and G. L. Peterson, “A function-to-task process model for adaptive automation system design,” *J. Hum. Comput. Stud.*, vol. 72, no. 12, pp. 822–834, 2014.

[14] D. B. Kaber, J. M. Riley, K.-W. Tan, and M. R. Endsley, “On the Design of Adaptive Automation for Complex Systems,” *Int. J. Cogn. Ergon.*, vol. 5, no. 1, pp. 37–57, Mar. 2001.

[15] T. Halverson, B. Reynolds, and L. Blaha, “SIMCog-JS : Simplified Interfacing for

Modeling Cognition - JavaScript.” 711th Human Performance Wing, Air Force Research Laboratory, pp. 39–44, 2013.

[16] M. Cohen, F. Ritter, and S. Haynes, “Herbal : A High-Level Language and Development Environment for Developing Cognitive Models in Soar,” in *Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation*, 2005, pp. 177–182.

[17] J. Anderson and M. Matessa, “An Overview of the EPIC Architecture for Cognition and Performance With Application to Human-Computer Interaction,” *Human-Computer Interact.*, vol. 12, no. 4, pp. 391–438, 1997.

[18] N. A. Giacobe, “A Picture Is Worth a Thousand Dollars,” *J. Cogn. Neurosci.*, vol. 21, no. 4, pp. 623–624, Apr. 2009.

[19] S. M. Galster and E. M. Johnson, “Sense-Assess-Augment: A Taxonomy for Human Effectiveness,” Wright-Patterson AFB, OH, 2013.

[20] A. Rowe, S. Spriggs, and D. Hooper, “Fusion: A Framework For Human Interaction With Flexible-Adaptive Automation Across Multiple Unmanned Systems,” *Int. Symp. Aircr. Psychol.*, vol. 18, no. 18th Annual, pp. 464–470, 2015.

[21] V. Finomore, a. Sitz, E. Blair, K. Rahill, M. Champion, G. Funke, V. Mancuso, and B. Knott, “Effects of Cyber Disruption in a Distributed Team Decision Making Task,” *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 57, no. 1, pp. 394–398, 2013.

[22] K. Sycara, A. Pannu, M. Williamson, D. Zeng, and K. Decker, “Distributed

intelligent agents," *IEEE Expert. Syst. their Appl.*, vol. 11, no. 6, pp. 36–46, 1996.

[23] S. Sarkar, A. C. Kak, and N. S. Nagaraja, "Metrics for analyzing module interactions in large software systems," *12th Asia-Pacific Softw. Eng. Conf. APSEC'05*, pp. 264–271, 2005.

[24] R. S. Pressman, "Chapter 5: Understanding Requirements," in *Software Engineering: A Practitioner's Approach*, 7th ed., McGraw-Hill, 2010, p. 119.

[25] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[26] H. Gomaa, *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*. Fairfax, Virginia: Cambridge University Press, 2011.

[27] M. A. Recarte, E. Pérez, A. Conchillo, and L. M. Nunes, "Mental Workload and Visual Impairment: Differences between Pupil, Blink, and Subjective Rating.," *Span. J. Psychol.*, vol. 11, no. 2, pp. 374–385, 2008.

[28] M. C. Dorneich, P. M. Ververs, S. Mathan, S. Whitlow, and C. C. Hayes, "Considering Etiquette in the Design of an Adaptive System," *J. Cogn. Eng. Decis. Mak.*, vol. 6, no. 2, pp. 243–265, 2012.

[29] M. Barnes, R. Parasuraman, and K. A. Cosenzo, "Adaptive Automation for Robotic Military Systems," in *RTO-TR-HFM-078 - Uninhabited Military Vehicles (UMVs): Human Factors Issues in Augmenting the Force*, NATO Science and Technology Organization, 2007, pp. 420–440.

[30] C. A. Russell and G. F. Wilson, "Application of Artificial Neural Networks for Air

Traffic Controller Functional State Classification," Jun. 2001.

[31] K. T. Durkee, S. M. Pappada, A. E. Ortiz, J. J. Feeney, and S. M. Galster, "System decision framework for augmenting human performance using real-time workload classifiers," in *2015 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision*, 2015, pp. 8–13.

[32] S. H. Fairclough and L. Venables, "Prediction of subjective states from psychophysiology: a multivariate approach.," *Biol. Psychol.*, vol. 71, no. 1, pp. 100–10, Jan. 2006.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
24-03-2016	Master's Thesis	August 2014 – March 2016

TITLE AND SUBTITLE

Analysis of Software Design Patterns in Human Cognitive Performance Experiments

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Roosma, Alexander C., Captain, USAF

5d. PROJECT NUMBER
JON 16G274

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-16-M-042

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Lead, Trust & Influence Program
Air Force Office of Scientific Research
875 N. Randolph Street, Arlington, VA 22203-1768
(703) 696-1142, ben.knott@us.af.mil
ATTN: Dr. Ben Knott

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR/RTC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRUBTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We present supporting research to the development of human-machine teaming through software design improvements to the execution of human cognitive performance experiments. This thesis sought to answer the following two research questions addressing the core functionality that these experiments rely on for execution and analysis. 1) What data infrastructure software requirements are necessary to execute the experimental design of human cognitive performance experiments? 2) How effectively does a central data mediator design pattern meet the time-alignment requirements of human cognitive performance studies? To answer these questions, this research contributes an exploration of establishing design patterns to reduce the cost of conducting human cognitive performance studies. The activities included in this exploration were a method for requirements gathering, a meta-study of recent experiments and a design pattern evaluation all focused on the experimental design domain.

15. SUBJECT TERMS

Software Design Patterns, Software Complexity, Cognitive Performance, Adaptive Automation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Brett Borghetti, AFIT/ENG
U	U	U	UU	130	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext 4612 (brett.borghetti@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18